PHYSICS BASED COMPRESSIVE SENSING FOR ADDITIVE MANUFACTURING PROCESS MONITORING

A Dissertation
Presented to
The Academic Faculty

By

Yanglong Lu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology
DECEMBER 2020

COPYRIGHT © 2020 BY YANGLONG LU

PHYSICS BASED COMPRESSIVE SENSING FOR ADDITIVE MANUFACTURING PROCESS MONITORING

Approved by:

Dr. Yan Wang, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Devesh Ranjan
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Alexander Alexeev
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Christopher Saldana
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Steven Liang
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Mark Davenport
School of Electrical & Computer Engineering
Georgia Institute of Technology

Date Approved: October 23, 2020

Acknowledgements

I would like to express my deepest gratitude to my advisor Dr. Yan Wang for all of his mentoring and guidance. He is very encouraging and let me explore my own research interests, and always willing to provide insightful advice when I encounter challenges in my research and study. He also helps me to improve my abilities in pursuing research career and I am motivated to think more critically. His professional and enthusiastic attitude has great impact on my research career. I also want to thank my committee members, Dr. Alexander Alexeev, Dr. Mark Davenport, Dr. Steven Liang, Dr. Devesh Ranjan and Dr. Christopher Saldana for their valuable suggestions to improve research results.

I really enjoy the research atmosphere in MSSE research group. I would like to thank each member in our group, Dehao Liu, Michael Kempner, Jesse Sestito, Luka Malashkhia, Dr. Ji-Hyeon Song and Dr. Anh Tran for the encouragement, feedback and assistance during my Ph.D. study. I also appreciate the helps from Dr. Jie Liu, Dr. Qi Zhou and Dr. Longchao Cao for providing experimental results and the guidance of experimental designs. I want to thank all the colleagues and friends I met in these years. Without their support, the completion of my dissertation would not be possible.

Finally, I would like to thank my family for their help and understanding through my entire life. As I have been away from home for more than eight years, their selfless love and concern always help me to keep a positive attitude even in the face of difficulties.

Table of Contents

List of Tables

List of Figures

Summary

Sensors play an important role in smart manufacturing. Different types of sensors have been used in process monitoring to ensure the quality of products. As a result, the life-cycle cost of quality control is rising. The reliability of sensors also affects the reliability of complex systems with a large number of sensors on-board. Another challenge is the available bandwidth in communication channels for transmission of large volumes of data. The original purpose of data cannot be fulfilled if they are not shared and used. In this research, a new approach that uses low-fidelity measurements with limited sensors to provide high-fidelity information in additive manufacturing (AM) process monitoring is investigated.

A physics-based compressive sensing (PBCS) approach is proposed to reduce the number of sensors and amount of data collection for AM process monitoring. PBCS significantly improves the compression ratio from traditional compressed sensing (CS) by incorporating the knowledge of physical phenomena in specific applications embodied as physics-based models. The general PBCS procedures include two steps. In the recovery step, an inverse problem is solved to recover the load vector in the discretized physical model based on the experimental measurements, usually obtained at the boundaries of domain. In the reconstruction step, the forward model is solved to predict the complete distributions of the physical quantities. PBCS has been demonstrated to monitor the temperature distribution in fused filament fabrication process. The performance of PBCS has been evaluated in monitoring the cooling process of the printed part when the printer head is paused and monitoring the real-time printing process. The systematic errors from PBCS reconstruction are also predicted and compensated based on a Gaussian process uncertainty quantification approach. PBCS has also been demonstrated to monitor the temperature and melt flow in selective laser melting for metal AM. Monitoring the melt pool with the size of a few

hundred micrometres in metal AM is difficult for conventional sensors due to limitations in spatial and temporal resolutions. Based on some temperature measurements on the top surface of melt pool and a thermofluid multi-physics model, PBCS can recover and reconstruct the complete temperature and velocity fields in three dimensions.

The sensing performance is further improved with a physics-constrained dictionary learning approach by optimizing the placement of low-fidelity measurements to obtain high-fidelity information. New dictionary learning algorithms are developed to optimize the basis matrix and measurement matrix in CS so that these matrices are customized to specific types of signals in one particular application to achieve the best accuracy and compression ratio. The new dictionary learning approach has been demonstrated with one-dimensional signals where sampling time stamps are optimized. In two-dimensional images, the optimal locations of pixels to sample are determined. When monitoring the surface temperature of the builds in AM processes with infrared thermal imaging systems, low-resolution pixel values at the designed locations can be used to recover high-resolution images. Based on the recovered images, more accurate three-dimensional temperature distributions can be reconstructed with PBCS.

With the application of PBCS in metal AM process monitoring, the three-dimensional temperature and velocity fields of the melt pool can be reconstructed with a limited number of measurements from low-cost sensors. Physics-constrained dictionary learning helps design sensor placement strategies. The proposed PBCS scheme provides a systematic and rigorous approach to design efficient sensing protocols for future manufacturing systems, where sensors are ubiquitously utilized in monitoring process and quality.

CHAPTER 1. Introduction

Additive manufacturing (AM) has become one of the most important techniques in recent decades, because it offers substantial design freedom for parts with complex geometries such as intricate internal features and lattice structures. For different AM processes, a fused filament fabrication (FFF) or material extrusion process is commonly used for polymers such as acrylonitrile butadiene styrene (ABS), polylactide (PLA), and Nylon. Metal AM processes including selective laser melting (SLM), electron beam melting (EBM), and direct energy deposition (DED) are used for metallic materials such as titanium alloys, stainless steel, and aluminium alloys. One major challenge of AM processes however is the variability of build qualities, where rapid melting and solidification are difficult to be monitored and controlled in these processes. In the FFF process, defects such as the detachment, the deformed object, the surface errors, and the deviation from the model can occur because of inappropriate thermal settings of machines and operational environments [1], which can result in abnormal melting and solidification rates and non-homogeneous temperature distributions of the builds. In metal AM processes, the microstructures of the solid build are largely influenced by the thermal history and melt flow during the rapid solidification process, where cooling rate, solute diffusion, liquid-solid and solid-solid phase transitions are affected. The grain shape, size, and orientation distributions, compositions and precipitates, as well as defects determine the physical properties of the solid build [2, 3, 4]. The thermal history and melt flow can be controlled by adjusting process parameters [5, 6, 7, 8] such as the power of heat source [9], scanning path [10], scanning speed [11, 12], and hatching space [13].

1.1 Challenges

1.1.1 Conventional sensors in monitoring the temperature field and melt flow in AM processes

Conventional thermal sensors such as thermocouples, pyrometers, and infrared (IR) thermal cameras have been applied to measure the temperature distribution in AM processes. However, the measurements are limited by temporal and spatial resolutions of the sensors. Especially in metal AM processes, temperature measurements of the melt pool with the size of micrometres have higher requirements of resolutions. Furthermore, these sensors only measure surface temperatures. The complete three-dimensional (3D) temperature distribution cannot be obtained. The measurement of melt flow in metal AM processes is even more challenging, where the liquid metal inside the small melt pool moves with a high velocity. High-speed high-energy x-ray imaging has been applied to measure fluid dynamics in the melt pool. However, the spatial resolution of the measured velocity field is limited by the density and spread of tracing micro-particles or bubbles in the melt pool. The operation of x-ray imaging systems is complex and costly.

Besides the above limitations of conventional sensors in monitoring the temperature field and melt flow in AM processes, there are three additional challenges for the applications of sensors in manufacturing systems in general. The first challenge is the life-cycle cost of sensors. The cost portion of sensing system installation, operation, and maintenance in the overall cost of manufacturing is rising. More importantly, the reliability of sensors will easily become the weakest link of the reliability of complex systems with a large number of sensors on-board. As a result, the maintenance cost of sensing subsystems is likely to be a major portion of system life-cycle costs. Undetected faulty sensors provide inaccurate information and can lead to costly wrong control actions. The second challenge is the bandwidth limitation of communication for the volume of data to be transmitted in an industry setting to enable remote monitoring, diagnostics, and control. Although sensor technologies will gradually become more affordable, the available bandwidth in

communication channels is always limited for transmission of large volumes of data collected by advanced sensors. If the data cannot be shared, processed, and used in real time, their original purpose for process monitoring will not be fulfilled. The third challenge is related to the physical limitations of data collection in practice, such as limited sensor accessibility in the actual manufacturing environment, limited space or limited numbers of sensors for senor placement, obstruction of the line of sight, and unobservable quantities of interest by sensors. Given the above limitations and challenges with in-situ monitoring of complex AM processes, there is a practical need to develop efficient and cost-effective sensing protocols.

1.1.2 Traditional compressed sensing

In the most recent decade, a new sampling and data collection approach, compressive sampling or compressed sensing (CS), was developed. CS is a new approach to capture and represent sparse signals with a reduction of sampling cost. With a small set of collected data samples, the original signal can be reconstructed by numerically solving an inverse problem. If signals can have a sparse representation in the reciprocal space through transformation, e.g. Fourier and wavelet transforms, the reconstruction can be fairly precise when the number of non-zero coefficients in the reciprocal space is small (i.e. *sparse*) and the transformation and projection operations are not correlated (i.e. *incoherent*). Traditional CS can reduce the cost associated with data collection. However, the compression is achieved purely by exploring the correlation within data collected by a sensor. This pure data-driven approach cannot achieve the compression ratio at a high level.

1.2 The proposed physics-based compressive sensing method

To address the challenges in monitoring AM processes discussed in Section 1.1, a physics-based compressive sensing (PBCS) approach is proposed. The PBCS framework is different from

traditional pure data-driven CS that was developed for generic signals. The PBCS relies on the domain knowledge of physics in applications. The recovery process in PBCS is to solve the inverse problem

$$\min\|\boldsymbol{y}^* - \boldsymbol{y}\|_2 \tag{1.1}$$

$$\text{subject to } \boldsymbol{y} = \boldsymbol{f}\left(t, \boldsymbol{\gamma}, \boldsymbol{d}, \dot{\boldsymbol{d}}, \ddot{\boldsymbol{d}}, \nabla\boldsymbol{d}, \dots\right) \tag{1.2}$$

$$\|\boldsymbol{\gamma}\|_0 < s_l \tag{1.3}$$

where the parameters $\boldsymbol{\gamma}$ of physical model $\boldsymbol{f}$ need to be recovered based on the predicted quantity $\boldsymbol{y}$ and measurements $\boldsymbol{y}^*$. s_l indicates the maximum number of non-zero values in the parameters $\boldsymbol{\gamma}$. The model describes the relationships among physical quantities $\boldsymbol{d}$ as well as their time and spatial derivatives ($\dot{\boldsymbol{d}}, \ddot{\boldsymbol{d}}, \nabla\boldsymbol{d}$, ...). The constraints in Eq.(1.2) are physical models, which are usually partial differential equations. The minimization can be based on the criteria of l_2 norm.

PBCS significantly improves the compression ratio from traditional CS by incorporating the knowledge of physical phenomena in specific applications embodied as physics-based models. The general PBCS procedures include two steps. In the recovery step, an inverse problem is solved to recover the load vector in the discretized physical model based on the experimental measurements, usually obtained at the boundaries of domain. In the reconstruction step, the forward model is solved to predict the complete distributions of the physical quantities.

The PBCS performance is further improved with a developed physics-constrained dictionary learning approach by optimizing the placement of low-fidelity measurements to obtain high-fidelity information. The measurement matrix and basis matrix in CS are optimized simultaneously to improve the compression ratio and recovery accuracy. With the physics-constrained dictionary learning approach, measurements required for PBCS are further reduced.

1.3 Technical contributions

1.3.1 A new sensing approach to efficiently monitor AM processes

A new sensing approach called physics-based compressive sensing is developed to efficiently monitor AM processes, which relies on the domain knowledge of physics in applications. Different from traditional CS techniques developed for generic one- or two-dimensional (2D) signals without the consideration of application domains, which are pure data-driven approaches, the PBCS approach significantly improves the compression ratio by one or two orders of magnitude based on the observation that the performance of compression is largely dependent on the inherent and domain-specific correlation within the collected data that is beyond statistical correlation. The PBCS formulations incorporated with a heat transfer model, a fluid flow model and a multi-physics model are developed to monitor the temperature distribution in the FFF process, the flow velocity field, and the temperature and melt flow of the melt pool in metal AM processes. The systematic error of PBCS is predicted and compensated based on a Gaussian process approach. The efficiency of the PBCS approach is further improved with a constrained orthogonal matching pursuit (OMP) algorithm and a domain decomposition approach.

1.3.2 A framework to optimize sensor placement

In traditional CS, the original signal can be reconstructed with a small set of collected data samples by solving an inverse problem. Dictionary learning methods were applied in combination with CS to improve the sparsity level of signals. However, they are limited in practical applications because the measurement matrix in CS is not designed explicitly. A physics-constrained dictionary learning approach is developed to optimize the placement of low-fidelity measurements to obtain high-fidelity information. The physics-constrained dictionary learning approaches are developed to optimize the basis matrix and measurement matrix in CS simultaneously so that these matrices are customized to specific types of signals in one particular application to achieve the best accuracy

and compression ratio. Additional physical constraints such as the coverage of measurements, the number of sensors, and sensor accessibility are considered in the learning process to improve the efficiency of data collection.

1.4 Dissertation organization

In the remainder of this dissertation, the background of CS, existing AM process monitoring techniques, inverse problems in heat transfer and computational fluid dynamics (CFD), and dictionary learning methods is described in CHAPTER 2. The PBCS framework to monitor the temperature distribution in the FFF process is demonstrated in CHAPTER 3. In CHAPTER 4, the PBCS mechanism is implemented to monitor the flow velocity field. In CHAPTER 5, the PBCS framework to monitor the temperature distribution and melt flow of the melt pool in metal AM processes is demonstrated. The physics-constrained dictionary learning approaches used to improve the efficiency of data collection for one-dimensional (1D) signals and 2D images are described in CHAPTER 6 and CHAPTER 7 respectively.

CHAPTER 2. Background

In this section, the background of traditional compressed sensing is introduced. The existing techniques for additive manufacturing process monitoring are described. Some of the existing work to solve inverse problems in heat transfer and computational fluid dynamics is also reviewed. The background of dictionary learning methods is also given.

2.1 Compressed sensing or compressive sampling

Compressed sensing or compressive sampling [14,15] was initially developed to solve the inverse problem of information recovery purely based on statistical characteristics of signals. Suppose that the original signal is represented in a discrete format as vector $\boldsymbol{s}$. It can be represented in the reciprocal space via transformation as $\boldsymbol{s} = \boldsymbol{\Psi\gamma}$ where $\boldsymbol{\Psi}$ is the matrix representation of transformation (or basis matrix) and $\boldsymbol{\gamma}$ is the vector of coefficients. The size of the original signal vector $\boldsymbol{s}$ is N. The size of the coefficients $\boldsymbol{\gamma}$ could be similar to N, however, only K of them are non-zero ($K<N$). That is, $\boldsymbol{\gamma}$ is K-sparse. When the signal is projected into another space to $\boldsymbol{y} = \boldsymbol{\Phi s}$ with a reduced dimension M ($M<N$) via a projection (or measurement) matrix $\boldsymbol{\Phi}$. The recovery of the original signal from the measured data is to solve the linear equations $\boldsymbol{y} = \boldsymbol{\Phi s} = \boldsymbol{\Phi\Psi\gamma} = \boldsymbol{\Theta\gamma}$. Loosely speaking, because of the K-sparsity, solving $\boldsymbol{\Theta\gamma} = \boldsymbol{y}$ first to find $\boldsymbol{\alpha}$ then reconstructing the original signal by $\boldsymbol{s} = \boldsymbol{\Psi\gamma}$ provides more accurate recovery than solving $\boldsymbol{\Phi s} = \boldsymbol{y}$ to find $\boldsymbol{s}$ directly. CS has been extensively applied to signal processing [16, 17], image processing [18,19,20], networked sensing [21], and others.

Recently, CS started being used to monitor machine health conditions. Chen et al. [22] used it to extract impulse components of roller bearing vibration signals. Wang et al. [23] applied to

time-frequency sparse representation of gear box vibration signals. Wang et al. [24] applied it to down sampling of bearing vibration signals. Tang et al. [25] classified the faults of rotating machinery with compressed measurements. Ding and He [26] applied to noise removal in the time-frequency domain. Yuan and Lu [27] applied CS to identify the health states of rolling bearing based on compressed vibration signals. Liu et al. [28] demonstrated the feasibility of using compressed features to identify rolling bearing states from acoustic emission signals.

All of the above approaches applied classical data-driven CS to machine condition monitoring. Signals were generally treated in the same way as any other type of data without the consideration of domain specific knowledge.

2.2 Additive manufacturing process monitoring

Additive manufacturing has great potentials in producing high-value low-volume products. However, the lack of quality assurance remains to be the major barrier for its engineering applications. Various sensing techniques have been developed to monitor AM processes.

In the FFF or material extrusion process, Dinwiddie *et al.* [29] used IR cameras to monitor the complete printing environment and heated extrusion head with a close-up view respectively. The temperature gradient of the part and the effect of different designs of extrusion heads were captured. Rao et al. [30] developed a heterogeneous sensor array including thermocouples, accelerometers, an IR camera and a real-time miniature video borescope to monitor the FFF process, and processed data from sensors with data-driven approaches. Baumann et al. [31] employed wireless sensors and a client-server system for the quality control of the printing process. Nuchitprasitchai *et al.* [32] designed single- and two-camera systems to detect a clogged nozzle, loss of filament, and incomplete prints. They also developed a 3D reconstruction algorithm from

two images captured by the two-camera system. Wu *et al.* [33, 34, 35, 36, 37] employed the acoustic emission technique to identify normal and abnormal states of machine conditions. Other techniques used in monitoring the FFF process include augmented reality technique [38], ultrasonic inspection technique [39], fibre Bragg grating sensor [40], and laser triangulation system [41].

For metal AM, optical imaging systems have been used to identify process failures such as the deformation of the part due to thermal stresses and overheating at overhang zones [42] and pores in the melt pool [43, 44]. High-speed optical camera can obtain the melt pool dynamics and melting front that moves at very high speeds [45]. Optical systems can provide observations with high temporal resolution. Yet the information obtained from optical systems is limited, because optical images can only be used to detect the contour of melt pool shapes at the top of build surface. Detail information such as the laser heat affected zone, melting and solidification conditions, and the spatial temperature gradient cannot be obtained. Thermal imaging systems have also been used to monitor metal AM processes. A bi-colour pyrometer integrated with an optical scanning system was developed by Pavlov et al. [46] to measure the surface temperature with a range between 900 °C and 2600 °C of the laser impact zone in the SLM process. The measurement was used to analyse the variation of the hatch distance and layer thickness. The spatial and temporal resolutions of the measured temperature profile was improved with a high-speed imaging thermography method developed by Hooper [47]. Dinwiddie et al. [48] used the thermal imaging system to visualize the porosity in overhang regions of the built part, which are bright features on the dark background. The temperature profile can also be used to observe the melting condition called balling phenomenon in the SLM process [49] and the flaw for the powder bed [50]. Different regions of the built part can be monitored by the inferred thermography such as the surface temperature of

the entire part [51, 52, 53], the heat affected zone which consists of the melt pool and the solidified region at elevated temperatures close to the melt pool [54, 55], and thermal profile of the melt pool [56, 57]. Compared to optical imaging systems, the measurements from thermal imaging systems can provide more information such as temperature gradients on the surface. It is challenging to monitor the fluid dynamics in the melt pool, because the size of the melt pool is in the scale of 100 μm and the flow patterns in the melt pool are complex. Some limited work has been done to measure fluid dynamics in the melt pool. Guo et al.[58, 59] used an in situ high-speed high-energy x-ray imaging system to measure the melt pool dynamics and quantify the velocity field by tracing tungsten micro-particles which are uniformly mixed with metal powders. The fluid dynamics in the melt pool are identified as different flow types based on flow directions. The measurement of fluid dynamics in the melt pool is also helpful to analyse the evolution of pores [60, 61] and the motion of metal powders [62] so that better build quality can be achieved.

Even though various techniques have been applied, they have limitations in monitoring AM processes. To monitor the temperature distribution in FFF and metal AM processes, the temporal and spatial resolutions of thermal sensors are usually low. Thermal sensors can only measure surface temperatures. However, the 3D temperature distribution cannot be obtained. To monitor the melt flow in metal AM processes, the velocity field can only be estimated indirectly by tracing micro-particles, which can introduce errors because micro-particles have different material properties from the build materials. The spatial resolution of velocity measurements based on particle tracing is low, because the number of micro-particles is limited.

2.3 Inverse problems in heat transfer and computational fluid dynamics

The proposed PBCS is to reconstruct temperature distributions and velocity fields from limited measurements by solving the inverse problem. Some limited efforts have been given to

study the inverse heat transfer problem [63], which is to estimate unknown quantities including boundary conditions of radiation [64] and convection [65,66], thermophysical properties, initial condition, source terms, and geometry [67] of a heated body with transient temperature measurements. Generic optimization techniques such as adjoint local search, conjugate gradient method [68], genetic algorithm [69] have been applied. The performance of these methods is sensitively dependent on the number of unknown parameters to be estimated. Excursion and oscillation of the solution may occur when the number of parameters is large.

There have also been studies of solving inverse problems in CFD modelling, which are to either optimize input parameters or reconstruct the velocity or pressure fields in the models. To optimize the input parameters, Xue et al. [70] combined the genetic algorithm (GA) with CFD to optimize the flow inlet conditions with measured wall temperatures. Inlet conditions including supply air temperature and velocities can be identified. GA was also applied to optimize the pressure distribution and airfoil geometries [71]. Inverse CFD has been applied to optimize the designs of pumps [72], IC engine ports and chambers [73], and turbine blades [74]. To reconstruct the velocity or pressure fields using the inverse approach, Waeytens et al. [75] applied the conjugate gradient method to reconstruct 2D flows by determining the optimal control velocities such as velocities near boundaries. Salloum et al. [76] applied traditional CS to reconstruct unstructured mesh datasets. Other techniques such as convolutional neural networks [77] and least square methods [78] were also used to reconstruct velocity fields. The inverse problems for the velocity field reconstruction were formulated to solve the issues of unknown effects of complex structures [79], and model calibration according to the measurements of skin friction [80] and surface pressure [81].

Different from the above work, the proposed PBCS relies on the sparsity of the coefficient

vector in the sense of CS to solve the inverse problem. If the vector of boundary and load conditions to be recovered has a high level of sparsity, CS can be very efficient and also provide very accurate recovery results. The algorithms for solving CS problems are generally more efficient and robust than other generic optimization methods such as GA and gradient-based algorithms used in the above approaches. Compared to traditional pure data-driven CS, the proposed PBCS approach utilizes the sparsity in the vector of boundary and load conditions for solving the inverse problem, which significantly improves the efficiency and robustness. The sparsity that can be identified from the physical models is mainly associated with the boundary conditions.

2.4 Dictionary learning

Various dictionary learning methods [82] have been developed to search for the sparsest representation of signals. The purpose is to find the optimal dictionary so that the sparsity is maximized for a specific type of signals. As a result, the original signals can be represented in a form of linear combinations of the learned dictionary and the sparse vector of coefficients. Some commonly used dictionary learning algorithms include the method of optimal directions (MOD) [83], K-SVD [84], the online dictionary learning [85] and others. The training process was also based on the maximum likelihood [86], least-square error [87, 88], and hidden Markov model [89].

Dictionary learning methods have been applied in combination with CS. For conventional CS, the basis matrix is usually predefined, so it is not directly related to the observed signals. Therefore, dictionary learning approaches have been developed to improve the sparsity level of the coefficient vector with a trained dictionary specifically based on the collected data. For example, Chen et al. [86] applied the dictionary learning method to improve the CS performance in extracting impulse components from noisy vibration signals. Lorintiu et al. [90] reconstructed ultrasound data with

CS and dictionary learning by K-SVD. It was shown that reconstruction errors are lower than conventional dictionaries based on Fourier or discrete cosine transformations. CS with learned dictionary was also applied for the reconstruction of magnetic resonance images [91, 92, 93, 94], videos [95] and electrocardiogram signals [96], and image denoising [97, 98, 99]. The existing dictionary learning approaches can improve the performance of CS. However, they are limited in practical applications because the measurement matrix is not designed explicitly. The measurement matrix is necessary to determine the locations of pixels to be measured and stored for 2D images or the time stamps of measurements for 1D signals.

Instead of learning the dictionary, which is the combination of the measurement matrix and the basis matrix, approaches to design the measurement and basis matrices separately were also developed. Duarte-Carvajalino and Sapiro [100] simultaneously optimized the measurement matrix and basis matrix with a new scheme called coupled-KSVD. The incoherence between the measurement and basis matrices was improved and resulted in better reconstruction performance. Bai et al. [101] further improved the framework with analytical solutions to update the measurement and basis matrices. It was shown that the convergence and accuracy of the solutions were improved for reconstruction of natural images. Nevertheless, in the above approaches, the optimized measurement matrix is dense. The dense measurement matrix cannot be used to determine the locations or time stamps of measurements or sampling in physical experiments. To be physically meaningful, measurement matrices should have only one non-zero entry in each row. The index of non-zero entry in each row indicates the time stamps to sample and store signals. Furthermore, physical constraints such as the data storage space, the number of measurements, sensor accessibility, and the energy consumption of data collection are important but considered in the existing approaches. Physical constraints ensure that the optimal performance is realizable

in practical applications.

The proposed physics-constrained dictionary learning framework optimizes the measurement and basis matrices simultaneously where the measurement matrix with only one non-zero entry in each row can directly indicate the time stamp of sampling. The physical constraint of the minimum sampling interval between stored and transmitted measurements is considered to reduce the redundancy for the storage and communication of temporally correlated data. The number of required measurements thus is optimized based on the physical constraints.

CHAPTER 3. Temperature Monitoring of Fused Filament Fabrication Process

3.1 Introduction

A PBCS approach is proposed to monitor the temperature distribution in the FFF process [102, 103], which is based on a heat transfer model. The PBCS formalism for the steady state and transient heat transfer problems is described in Section 3.2. After finite-element discretization of the heat transfer model, the heat load vector is first recovered with a developed constrained OMP algorithm. The constrained OMP algorithm can predict non-zero coefficient indices in the recovered sparse vector by using physical knowledge as the additional constraint, which can improve the recovery performance. With the recovered heat load vector, the complete temperature distribution can be reconstructed by solving the forward heat transfer model. The proposed PBCS approach is used to monitor the steady state temperature distribution for both of 2D and 3D cases in Section 3.3. Comparisons of different recovery algorithms and sensing strategies such as single-probe measurement and low-fidelity measurement are performed. The transient temperature distribution is monitored with PBCS in Section 3.4. The cooling process of the printed part when the printer head is paused is monitored. A domain decomposition scheme is developed to reduce the computational cost in the recovery process. With the domain decomposition scheme, the size of the basis and measurement matrices in PBCS recovery and reconstruction can be significantly reduced, which makes PBCS more suitable to monitor large domains or complex systems. The systematic errors from PBCS reconstruction are also predicted and compensated based on a Gaussian process uncertainty quantification approach. PBCS for real-time model update is demonstrated in Section 3.5. The geometry changes in the printing process are considered. The domain decomposition scheme is also applied in the real-time process monitoring.

3.2 PBCS for temperature monitoring

3.2.1 Finite-element discretization of heat transfer model

The time-dependent heat transfer problem is modelled by a partial differential equation (PDE) as

$$\rho c_p \dot{T} - \kappa \Delta T = 0 \quad \text{in } \Omega \tag{3.1}$$

where ρ is the density, c_p is the specific heat at constant pressure, κ is the thermal conductivity, T is the temperature and Ω indicates the modeling domain. If boundary conditions such as heat flux and convection are applied in the subdomain $\partial\Omega$, a balance of energy transferred across the boundary can be expressed as

$$\kappa \nabla T \cdot \hat{n} + h_c(T - T_\infty) = g \ \text{ in } \partial\Omega \tag{3.2}$$

where h_c is the heat transfer coefficient for thermal convection, T_∞ is the ambient temperature, g is the heat flux, and $\hat{n}$ donates the unit normal vector to the boundary.

By introducing a shape function θ for the vector of nodal temperatures $\boldsymbol{T}$, the variable T can be approximated as

$$T = \theta^T \boldsymbol{T} \tag{3.3}$$

With test function w, the weak formulation of Eq. (3.1) over an element domain Ω^e including the boundary condition in Eq. (3.2) is

$$\int_{\Omega^e} \left(w\rho c_p \dot{T} + \kappa \nabla \mathrm{w} \nabla T \right) d\Omega^e - \int_{\partial\Omega^e} (w\kappa \nabla T \cdot \hat{n}\,) d(\partial\Omega^e) = 0 \tag{3.4}$$

Test function w can be replaced with θ. By substituting Eq. (3.3) into Eq. (3.4), finite element equations over an element domain Ω^e can be represented as

$$\left[\int_{\Omega^e} \rho c_p \theta \theta^T d\Omega^e \right] \dot{\boldsymbol{T}} + \left[\int_{\Omega^e} \kappa \frac{\partial \theta}{\partial x_i} \frac{\partial \theta^T}{\partial x_j} d\Omega^e \right] \boldsymbol{T} + \left[\int_{\Omega^e} h_c \theta \theta^T d\Omega^e \right] \boldsymbol{T} = \int_{\Omega^e} (g + h_c T_\infty) \theta d\Omega^e \tag{3.5}$$

Eq. (3.5) can be reformulated as

$$\mathbf{M}\dot{\boldsymbol{T}} + \mathbf{K}\boldsymbol{T} = \boldsymbol{L} \tag{3.6}$$

where the coefficient matrices are defined as

$$\begin{aligned}\mathbf{M} &= \int_{\Omega^{\mathrm{e}}} \rho c_p \theta\theta^T d\Omega^{\mathrm{e}} \\ \mathbf{K} &= \int_{\Omega^{\mathrm{e}}} \kappa \frac{\partial\theta}{\partial x_i}\frac{\partial\theta^T}{\partial x_j} d\Omega^{\mathrm{e}} + \left[\int_{\Omega^{\mathrm{e}}} h_c \theta\theta^T d\Omega^{\mathrm{e}}\right] \\ \boldsymbol{L} &= \int_{\Omega^{\mathrm{e}}} (g + h_c T_\infty)\theta d\Omega^{\mathrm{e}}\end{aligned} \tag{3.7}$$

The time discretization of PDE in Eq.(3.6) is based on the backward Euler discretization, as

$$\mathbf{M}\frac{\boldsymbol{T}_n - \boldsymbol{T}_{n-1}}{\Delta t} + \mathbf{K}\boldsymbol{T}_{n-1} = \boldsymbol{L} \tag{3.8}$$

where Δt is the time step. In the dynamic process where the temperature field changes, the temperature $\boldsymbol{T}_n$ at current time step n can be computed from the previous time step $n-1$ by

$$\boldsymbol{T}_n = \boldsymbol{\alpha}\boldsymbol{T}_{n-1} + \boldsymbol{\beta}\boldsymbol{L} \tag{3.9}$$

or recursively from the initial temperature $\boldsymbol{T}_0$ as

$$\boldsymbol{T}_n = \boldsymbol{\alpha}^n\boldsymbol{T}_0 + (\boldsymbol{\alpha}^{n-1}\boldsymbol{\beta} + \boldsymbol{\alpha}^{n-2}\boldsymbol{\beta} + \cdots + \boldsymbol{\alpha}\boldsymbol{\beta} + \boldsymbol{\beta})\boldsymbol{L} \tag{3.10}$$

where $\boldsymbol{\alpha} = (\mathbf{M} + \Delta t\mathbf{K})^{-1} \cdot \mathbf{M}$ and $\boldsymbol{\beta} = (\mathbf{M} + \Delta t\mathbf{K})^{-1}\Delta t$,

3.2.2 PBCS for steady state problem

In the PBCS process for the steady state problem, Eq. (3.6) is simplified as

$$\boldsymbol{T} = \mathbf{K}^{-1}\boldsymbol{L} \tag{3.11}$$

The heat load vector $\boldsymbol{L}$ in Eq. (3.11) is first recovered from a few measurements. The measured temperature $\boldsymbol{T}^*$ is obtained or sampled based on the measurement matrix $\boldsymbol{\Phi}$.

The measurement matrix $\boldsymbol{\Phi}$ is constructed to indicate locations of the limited measurements.

An example of the measurement matrix is

$$\boldsymbol{\Phi} = \begin{bmatrix} 0 & 1 & 0 & & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{3.12}$$

Here, the column index where the value of 1 is located in each row indicates the location of each sensor measurement corresponding to the node in the discretized Temperature $\boldsymbol{T}$. $\boldsymbol{\Phi}$ has the size of $M \times N$, where M is the number of measurements and N is the total number of nodes in the complete discretized field, and $M \ll N$. PBCS to monitor the steady state temperature distribution is to solve

$$\min \|\boldsymbol{T}^* - \boldsymbol{\Phi T}\|_2 \tag{3.13}$$

$$\text{subject to } \boldsymbol{T} = \mathbf{K^{-1}} \boldsymbol{L} \tag{3.14}$$

$$\|\boldsymbol{L}\|_0 < s_l \tag{3.15}$$

where s_l is the minimum sparsity level of $\boldsymbol{L}$. The heat load vector $\boldsymbol{L}$ can be recovered from the measurements $\boldsymbol{T}^*$, which is a small portion of the complete temperature field $\boldsymbol{T}$. Then the temperature field can be reconstructed as $\boldsymbol{T} = \mathbf{K^{-1}} \boldsymbol{L}$.

3.2.3 PBCS for transient problem

For the transient problem, based on Eq. (3.10), a first-order system for *n* time steps is obtained as

$$\begin{bmatrix} \boldsymbol{T}_1 - \boldsymbol{\alpha T}_0 \\ \boldsymbol{T}_2 - \boldsymbol{\alpha}^2 \boldsymbol{T}_0 \\ \vdots \\ \boldsymbol{T}_n - \boldsymbol{\alpha}^n \boldsymbol{T}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\alpha\beta} + \boldsymbol{\beta} \\ \vdots \\ \boldsymbol{\alpha}^{n-1}\boldsymbol{\beta} + \boldsymbol{\alpha}^{n-2}\boldsymbol{\beta} + \cdots + \boldsymbol{\alpha\beta} + \boldsymbol{\beta} \end{bmatrix} \boldsymbol{L} \tag{3.16}$$

The measurement matrix $\boldsymbol{\Phi}$ for *n* time steps is

$$\boldsymbol{\Phi} = \begin{bmatrix} [\boldsymbol{\Phi}_1] & 0 & \cdots & 0 \\ 0 & [\boldsymbol{\Phi}_2] & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & [\boldsymbol{\Phi}_n] \end{bmatrix} \tag{3.17}$$

where $[\boldsymbol{\Phi}_j]$ indicates the locations of measurements at the j^{th} time step. If only the temperature at the final time step is measured and used to reconstruct previous temperature fields, then $\boldsymbol{\Phi}$ can be simplified as $Diag([0 \quad 0 \quad ... \quad \boldsymbol{\Phi}_n])$. If only the temperature at the first time step is measured and used to predict future temperature fields, then $\boldsymbol{\Phi}$ will be $Diag([\boldsymbol{\Phi}_1 \quad 0 \quad ... \quad 0])$.

In the case that the temperature at the final time step is measured, the PBCS recovery is to find the load vector $\boldsymbol{L}$ such that the difference between the experimental measurements $\boldsymbol{T}^*$ at the n^{th} time step and the selected predictions $\widetilde{\boldsymbol{T}}_n = \boldsymbol{\Phi}_n T_n$ from model is minimized by solving

$$\min \left\| \boldsymbol{T}^* - \widetilde{\boldsymbol{T}}_n \right\|_2 \tag{3.18}$$

$$\text{subject to } (\boldsymbol{T}_n - \boldsymbol{\alpha}^n \boldsymbol{T}_0) = (\boldsymbol{\alpha}^{n-1}\boldsymbol{\beta} + \boldsymbol{\alpha}^{n-2}\boldsymbol{\beta} + \cdots + \boldsymbol{\alpha}\boldsymbol{\beta} + \boldsymbol{\beta})\boldsymbol{L} \tag{3.19}$$

$$\|\boldsymbol{L}\|_0 < s_l \tag{3.20}$$

The load vector $\boldsymbol{L}$ is recovered with a few measurements at the final time step, and assumed to be constant during these time steps. With the recovered load vector $\boldsymbol{L}$, the complete temperature fields along time can be reconstructed with Eq. (3.16).

3.2.4 Constrained OMP

Orthogonal matching pursuit (OMP) [104] has been widely used in signal recovery problems. In the OMP algorithm, non-zero coefficient indices in the recovered sparse vector need to be determined. The incoherence between the measurement matrix and the basis matrix is required to successfully perform OMP. When significant coherence exists, the prediction of non-zero coefficient indices can be wrong. However, the requirement of incoherence is difficult to meet in many real-world problems. To deal with this challenge, some improvements have been done by

either modifying the measurement matrix [105], or modified algorithms such as BOMP [106], generalized OMP [107] and subspace pursuit [108].

Here, a modified OMP, called constrained OMP [109], is proposed to predict non-zero coefficient indices by using physical knowledge as the additional constraint, which is especially useful to improve the performance of PBCS. In PBCS for temperature monitoring, the physical model is built based on the heat transfer equation. There is no volumetric heat source applied. Therefore, the indices of non-zero values in the heat load vector $\boldsymbol{L}$ are known in advance, which should be in the index set of boundaries. Therefore, the predicted non-zero coefficient indices with constraint OMP must belong to the boundary nodal indices. The algorithm of constrained OMP is listed in Table 1.

Table 1. Constrained OMP algorithm

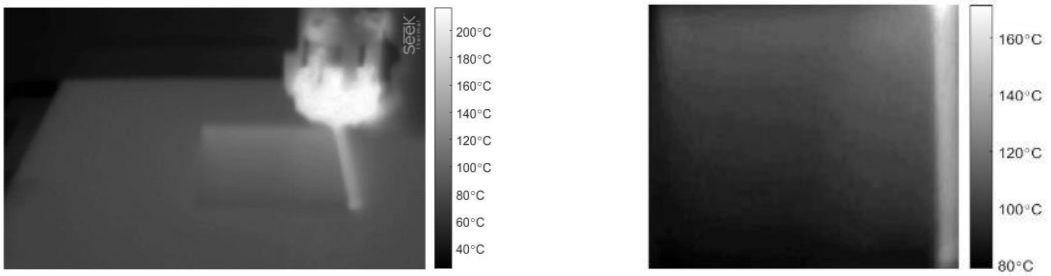
Input:	Measurements $\boldsymbol{T}^*$; measurement matrix $\boldsymbol{\Phi}$; model matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. (or combined intrinsic matrix $\boldsymbol{\Theta}$); desired sparsity k.
Output:	Estimated load vector $\boldsymbol{L}$.
	1. Initialize the residual $\boldsymbol{r_0} = \boldsymbol{T}^*$, the index set $\Lambda_0 = \emptyset$, iteration counter $i = 0$, and intrinsic matrix $\boldsymbol{\Theta}_0 = \emptyset$. 2. Find the index λ_i corresponding to the largest magnitude entry in $\boldsymbol{\Theta}^{\mathbf{T}}\boldsymbol{r}_{i-1}$ 3. If λ_i belongs to measurement locations in $\boldsymbol{\Phi}$, then go to Step 4. Otherwise, set λ_i^{th} column of $\boldsymbol{\Theta}$ to zero and repeat Step 2. 4. Update the index set $\Lambda_i = \Lambda_{i-1} \cup \{\lambda_i\}$ and intrinsic matrix $\boldsymbol{\Theta}_i = [\boldsymbol{\Theta}_{i-1} \ \ \boldsymbol{\varphi}_{\lambda_i}]$, where $\boldsymbol{\varphi}_{\lambda_i}$ is the λ_i^{th} column of the original intrinsic matrix $\boldsymbol{\Theta}$. 5. Find the new vector by solving $\boldsymbol{L}_i = \text{argmin}_{\boldsymbol{L}} \|\boldsymbol{T}^* - \boldsymbol{\Theta}_i \boldsymbol{L}\|_2$ 6. Compute new measurement vector and residual as $\boldsymbol{T}_i^* = \boldsymbol{\Theta}_i \boldsymbol{L}_i \quad \boldsymbol{r}_i = \boldsymbol{T}^* - \boldsymbol{T}_i^*$ 7. $i = i + 1$ and return to Step 2 if $i < k$

3.3 Demonstration of PBCS for the steady state problem

3.3.1 Physical Experiments

A Hyrel 3D printer was used in the experiment to print a simple box with the size of 45mm×45mm×6mm. The material was acrylonitrile butadiene styrene (ABS). The typical approach to monitor the temperature distribution is using thermal imaging systems. During the printing process, a Seek thermal camera was used to capture the gray-scale image of the temperature field, which is shown in Figure 1(a). Particularly, the temperature distribution at the top surface of the print is the domain of interest and is used to assess the PBCS accuracy.

Since the view angle of the experimentally captured image for the domain is different from a rectangular 2D image, an image registration process was performed to map the domain to a regular image. The right edge of the domain is the newly printed line segment with the highest temperature values. Thus, the right edge was used as a reference feature in image registration. The image registration tool in Matlab was used. Affine transformations including translation, rotation, scaling, and shearing were applied to the experimentally captured image. After image registration, the image was scaled to 45×45 pixels. Each pixel in the gray-scale image was converted to a temperature scale with a linear map, where the temperature was derived from the pixel value with a linear interpolation between the minimum and maximum temperatures and rounded to the nearest integer. The processed image with each pixel value as the actual temperature is shown in Figure 1(b). The temperature distribution on the top surface of the printed part is used to compare with PBCS reconstruction in Section 3.3.2.2.



(a) (b)

Figure 1. Thermal image captured with the Seek thermal camera. (a) Original thermal image; (b) Processed image after converting pixel values to temperatures

3.3.2 Results and discussion

In this section, a simple 2D thermal model of the material extrusion process is first used to illustrate the proposed PBCS approach in Section 3.3.2.1. Sensitivities of measurement strategies are also analysed. Then a PBCS based 3D temperature distribution monitoring is used to demonstrate the new sensing method in Sections 3.3.2.2.

3.3.2.1 2D thermal model

A 2D physical model of material extrusion is constructed, as illustrated in Figure 2, where one quarter of the printing area is modelled. The extruder as the heat source is located at the bottom left corner of the domain without movement, and the top and right boundaries correspond to the hotbed temperature. 88 quadratic triangular elements are used in the discretized finite-element formulation, with a total of 205 nodes in the 2D domain. Following the regular finite element modelling, a heat load vector is assigned, and the temperature distribution predicted from the model is used as the reference for comparison. During PBCS, some nodal temperatures are selected and treated as if they were measurements. They are then used to recover the heat load vector. The recovered heat load vector is employed to reconstruct the temperature distribution, which is compared with the original one. The purpose of not taking measurements directly from the actual

physical experiments is to illustrate the PBCS error associated with recovery algorithms without the confounding effect of measurement errors in physical experiments. Two cases are studied. In the first case, only temperatures of the extruder and the hotbed are sampled, denoted by circles and triangles respectively. In the second case, temperature samples are taken at the boundary and a few locations inside the domain.

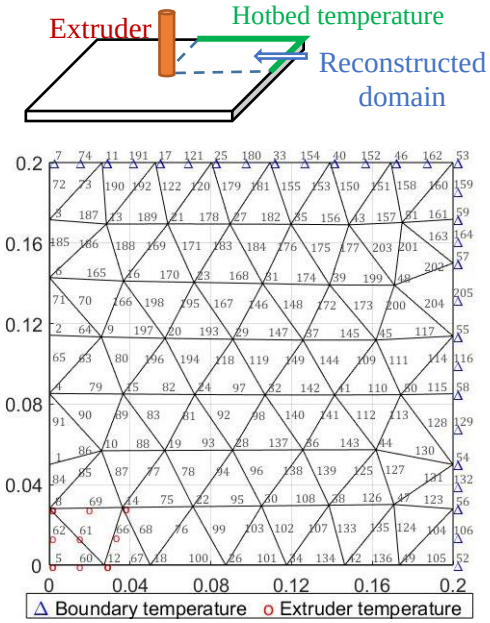
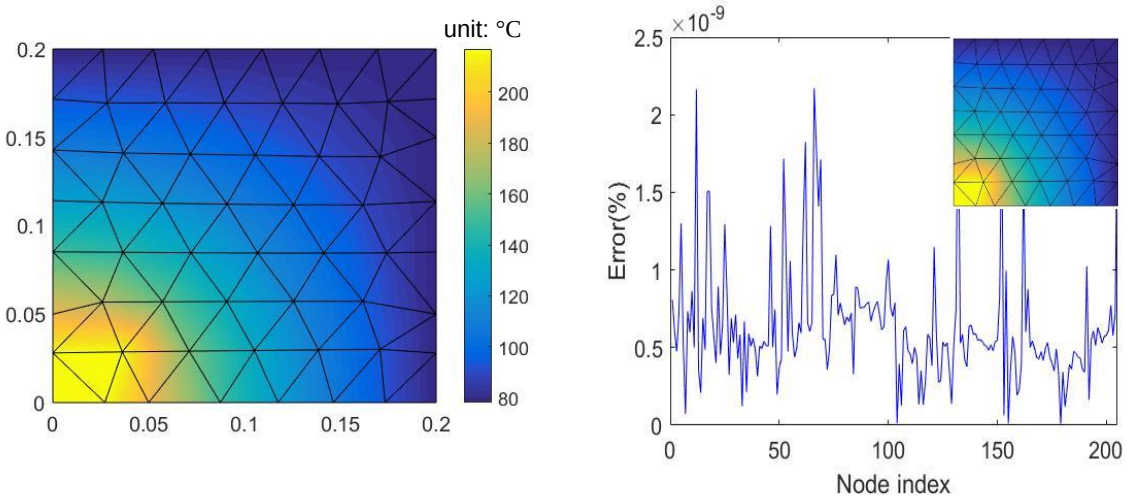


Figure 2. 2D finite-element domain of temperature distribution in °C for reconstruction

Case 1: Single-probe measurements at extruder and hotbed

Fixed temperature boundary conditions are applied to the top- and right-side of the domain (hotbed) and the bottom-left corner (extruder). It is reasonable to assume that all 29 nodes at the top- and right-side boundaries have the same temperature of hotbed, which is 78°C, whereas 9 nodes on the bottom-left corner have the same temperature of extruder, which is 217°C. The sparse heat load vector $\boldsymbol{L}$ is first recovered from temperatures at these 38 nodes**.** Then all nodal temperatures $\boldsymbol{T}$ in Eq.(3.11) are reconstructed. After the heat load vector is recovered, those values in the vector that are smaller than a threshold of 10^{-5} are set to be zeros to eliminate the numerical

round-off effect. The PBCS reconstructed temperature is the same as the original one from the finite element model in Figure 3(a). The differences between temperatures of all 205 nodes based on the basis pursuit algorithm [110] are shown in Figure 3(b). The reconstructions are exact, and the heat load vector can be lossly recovered.



(a) (b)

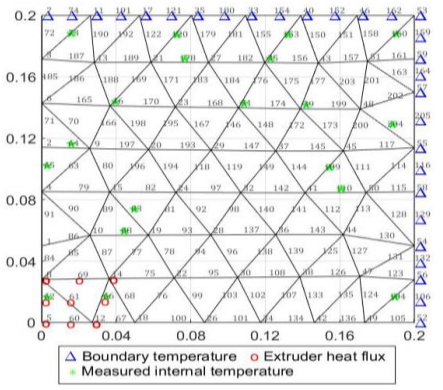
Figure 3. Reconstruction of 2D temperature distribution. (a) Original temperature distribution from the finite-element model; (b) Reconstruction error at the all nodal positions with the basis pursuit algorithm.

Case 2: Low-fidelity measurement inside printing domain and hotbed

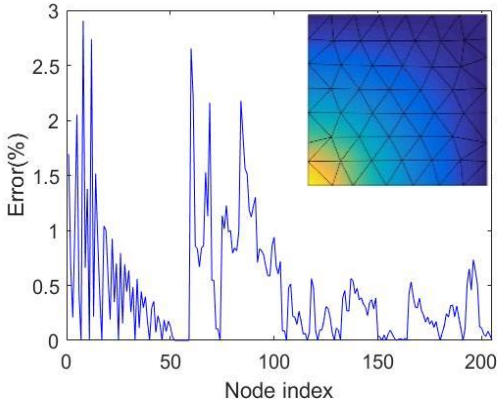
The measurement matrix $\boldsymbol{\Phi}$ in Eq. (3.14) contains the indices of nodes in the model, which indicate the locations where the temperatures need to be measured. Choosing different locations of measurements may result in different levels of reconstruction accuracy.

In the measurement strategy shown in Figure 4(a), instead of measuring the extruder temperature, some internal temperatures within the domain are used and the locations are highlighted with stars (*). This strategy can be regarded as measurements from pyrometers at various locations. The hotbed temperature is also used for reconstruction. A total of 48 nodal temperatures, including 19 internal nodes and 29 nodes at the top- and right-side boundaries, are used for recovery. The boundary condition of the physical model is changed with heat flux at

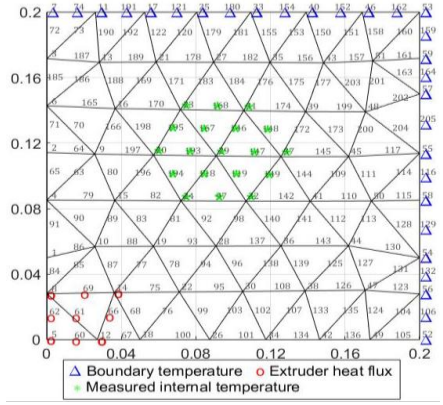
extruder nodes. Figure 4(b) shows the reconstruction results and errors. The errors are larger than the ones in Figure 3(b). If the internal measurements are concentrated in a local region, which can be regarded as the case where the infrared camera measures a portion of the domain, as shown in Figure 4(c), the reconstruction results are different, as shown in Figure 4(d), where errors further increase from the ones in Figure 4(b). When single-probe measurement is used to measure the internal temperatures and only one temperature reading is taken for all 19 internal nodes in Figure 4(c), the reconstruction errors shown in Figure 4(e) are close to the previous ones in Figure 4(d) where the low-fidelity measurement was taken and different nodal values were used in reconstruction, because the temperature gradient within this measured region is small.



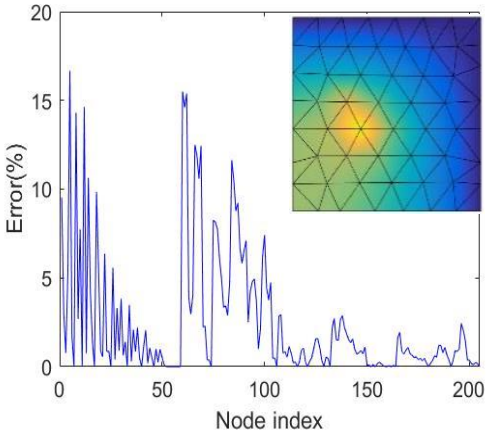
(a)



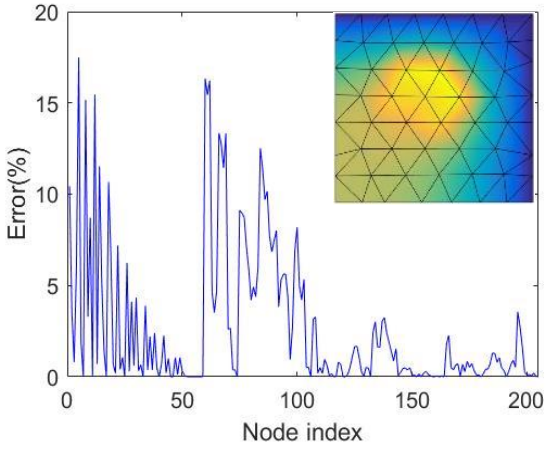
(b)



(c)



(d)



(e)

Figure 4. Effects of measurement strategies. (a) Scattered internal temperature measurements in °C; (b) Reconstruction errors from (a) at the nodal positions with the basis pursuit algorithm; (c) Concentrated internal temperature measurements in °C; (d) Reconstruction errors from (c) at the nodal positions with the basis pursuit algorithm; (e) Reconstruction errors when the single-probe measurement is taken for internal temperatures instead in (c) with the basis pursuit algorithm.

In classical CS, depending on the reconstruction algorithms, the minimum number of measurements is in an order between $K\log(N/K)$ and K, which is associated with the level of sparsity K. In PBCS, the number of measurements can be reduced based on a prior knowledge of the physical system. For instance, in the example in Figure 3, the number of measurements can be reduced to only two single-probe measurements, i.e. hotbed and extruder. Based on the knowledge of the system to be modelled, multiple nodes can be assigned to have the same temperature value. If the location of the extruder is not available and only the physical model is applied in PBCS, random measurements can be used to reconstruct the complete measurement domain as shown in Figure 4(a). The physics-based approach thus reduces the number of sensors to be deployed. Nevertheless, the strategy of reducing the number of sensors and designing locations of measurements could affect the reconstruction results.

3.3.2.2 Monitoring 3D thermal distribution

Here, the PBCS approach to monitor the printing process corresponding to the first physical experiment described in Section 3.3.1 is demonstrated. Figure 5 shows a 3D model of the printed part, where four newly printed lines form a separate segment attached on the top left of the part. The dimension of each printed line is 0.75mm×45mm×1mm. The extruder is currently at the location (3, 0, 6). Convection boundary conditions are applied to faces F1 to F4 and F6. Heat flux from the hotbed goes through face F5. Conduction matrix **K** in Eq.(3.11) is generated with $h_c = 25\ \mathrm{W \cdot m^{-2} \cdot K}$ and $\kappa = 0.1\ \mathrm{W \cdot m^{-2} \cdot K}$. The density of the material is 1.04 $\mathrm{g \cdot cm^{-3}}$, and the heat capacity is 1420 $\mathrm{J \cdot kg^{-1} \cdot K}$. A 3D mesh model is generated and the maximum mesh size is 8 mm, which is the length of the longest edge in the quadratic tetrahedral element. There are a total of 346 elements and 787 nodes.

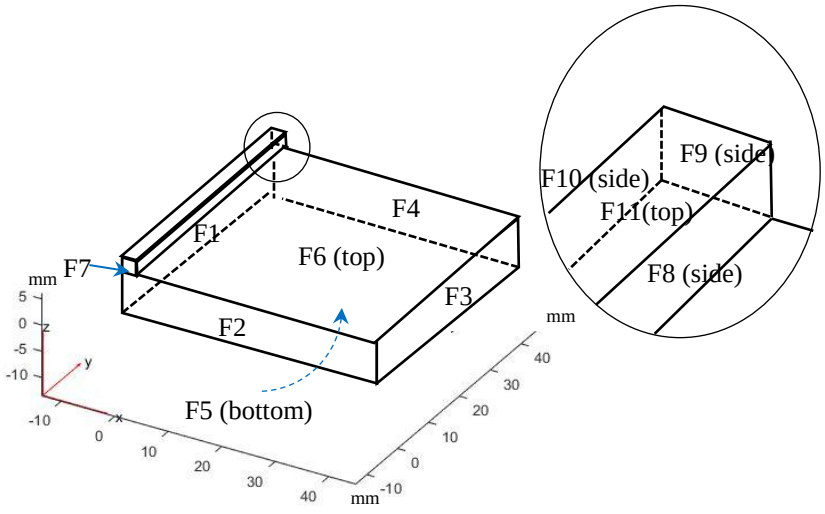


Figure 5. The printing domain in material extrusion process

In the first example, the PBCS reconstruction is based on the single-probe measurement, where one temperature reading on each of side faces F1 to F4 and top face of the newly printed segment F11 is taken to reconstruct the complete 3D temperature distribution. From the experimental measurement, 127°C, 103°C, 88°C, and 82°C are the temperature readings sampled at the centre of each boundary edge from the thermal image in Figure 1(b), which are labelled by dots in Figure 6(a). They are assigned as the temperatures of all nodes on the side faces F1 to F4 respectively in the physical model. 140°C is measured at the centre of the top face formed by the newly printed segment and assigned as the nodal temperatures of face F11. The reconstructed 3D temperature distribution is shown in Figure 6(b). Since the true 3D temperature distribution cannot be measured directly, the error associated with the PBCS reconstruction is unknown. To have an approximated estimation of the reconstruction error from the single-probe measurement, a baseline reconstruction is also performed, where the 2D temperature distribution on the top faces F6 and F11 from the experiment in Figure 6(a) is used to assign the nodal temperatures on these two faces

in the model for reconstruction. In other words, the nodes on these two faces take the actual temperatures respectively. The reconstructed baseline 3D temperature distribution is shown in Figure 6(c). The temperature differences of all 787 nodes between PBCS reconstruction with the single-probe measurement and the baseline 3D distribution are shown in Figure 6(d). The average error indicated by the dash line is 5.78% and the standard deviation of errors is 6.41%. To some extent the differences indicate the PBCS reconstruction error, although the ground truth is unknown. In the single-probe measurement, the compression ratio is 787/5=157.4. Here, OMP algorithm [104] is used for recovery. Our test also showed that the compression ratio is about 3 if classical CS is used to recover those nodal temperatures with the level of sparsity in the thermal load vector.

To compare the PBCS reconstruction with the direct full measurements of 2D temperature distribution from the thermal camera, the nodal temperatures on the top surfaces F6 and F11 from PBCS prediction are extracted. A 2D linear interpolation based on the nodal temperatures on F6 and F11 is used to store the temperature distribution as a matrix similar to an image. The size of the interpolated image from PBCS reconstruction is 45×45 pixels, which has the same size as the directly measured image in Figure 6(a). The temperatures between the two images then are compared pixel by pixel. The differences are the PBCS sensing errors and are plotted with respect to x and y coordinates of pixels in Figure 6(e). The average error is 6.86% and the standard deviation is 7.03%. Most errors come from face F11, where the temperature gradient is high in this small region. Assigning one temperature value to all nodes is not a good approximation. Adding more measurements can improve accuracy further. Neverthess, with the one temperature on F11, the reconstructed temperature distribution on face F6 is fair. The average and standard deviation of errors excluding F11 are 5.72% and 5.02%. Note that the whole domain of print is of interest

because the complete thermal history is important to monitor the materials' phase transformation process.

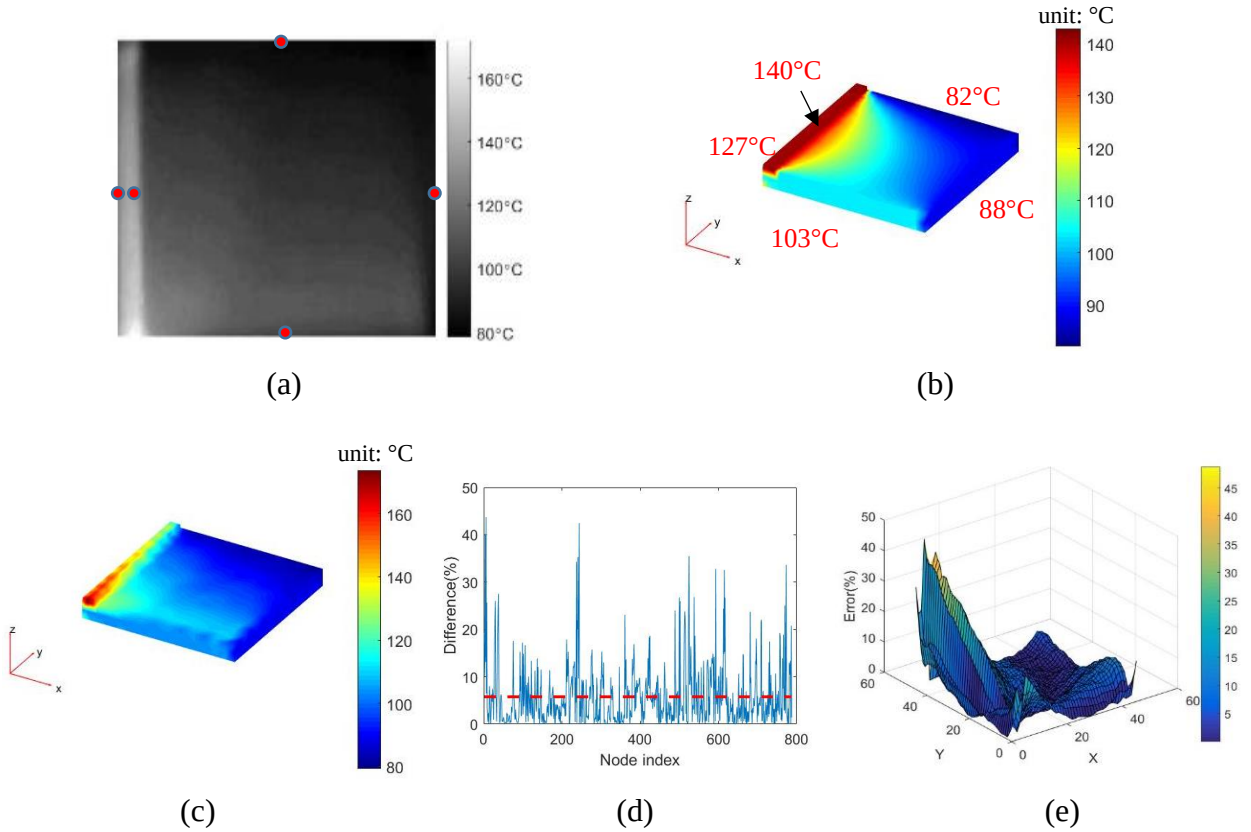


Figure 6. PBCS reconstruction of 3D temperature distribution from the single-probe measurement. (a) 2D temperature field of top surface from the experimental measurement; (b) Reconstructed 3D temperature distribution from four side surface readings and one reading on face F11; (c) Baseline 3D reconstruction based on top surface temperature distributions and four side surface temperatures; (d) Temperature differences of all nodes between reconstruction from the single-probe measurement and the baseline reconstruction; and (e) Pixel-by-pixel differences between PBCS reconstructed top surface temperature distribution and the direct measurement from camera.

A second example is to illustrate that more experimental measurements can reduce the reconstruction error. Based on the first example of the single-probe measurement, more temperature readings are taken from faces F6 and F11. The reconstructed 3D temperature distribution can be closer to the baseline reconstruction. The 2D domain of faces F6 and F11 is

divided into several regions along *x*-direction, as shown in Figure 7(a). The region enclosed by a box is the newly printed segment as face F11, which has much higher temperatures than other regions. Face F6 is further subdivided into different regions. One temperature reading at the centre of bottom and top edges for each region is taken and is assigned to all nodes on bottom and top edges. Nodal temperatures along *y*-direction are then assigned by linearly interpolating temperatures on edges. In Figure 7(b), the reconstructed 3D temperature distribution is based on four temperature readings from faces F1 to F4, two readings from face F11, and two readings from face F6 without further subdivision. The average nodal temperature difference between the reconstruction with a total of 8 measurements and the baseline is 5.68%, and the standard deviation is 5.64%, as shown in Figure 7(c). It is seen that the errors are reduced from the ones in Figure 6(d). The compression ratio for 8 measurement readings is 787/8=98.38. In Figure 7(d-e), measurements include four temperature readings from faces F1 to F4, two readings from face F11, and twelve readings from face F6, where face F6 is further divided into six regions. In Figure 7(f-g), a total of ninety readings are taken for reconstruction, including eighty-four readings from faces F6, where face F6 is divided into forty-two regions. The average differences are 4.39% and 3.94% respectively and the corresponding standard deviations are 4.26% and 3.48%.

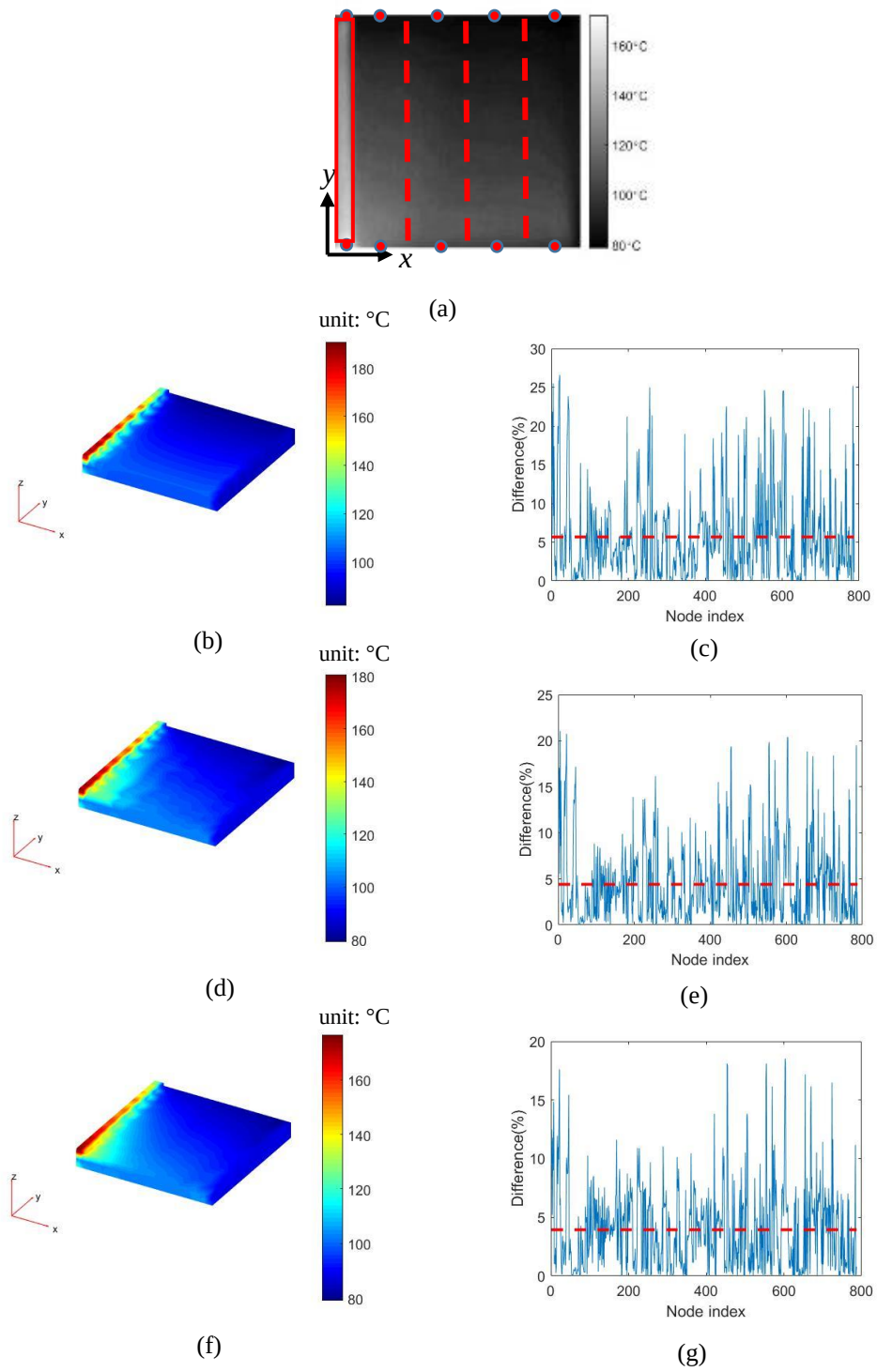


Figure 7. Nodal temperature errors associated with the PBCS reconstructions are reduced when more measurements are taken. (a) Top faces F6 and F11 are subdivided into regions with two readings in each; (b) Reconstruction from 4 readings on faces F1 to F4 and 4 on top faces F6 and F11, and (c) the nodal temperature errors; (d) Reconstruction from 4 readings on faces F1 to F4 and 14 on top faces F6 and F11, and (e) the nodal temperature errors; (f) Reconstruction from 4 readings on faces F1 to F4 and 86 on top faces F6 and F11, and (g) the nodal temperature errors.

3.4 Demonstration of PBCS for the transient problem

3.4.1 Domain decomposition

In PBCS, the formation of matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ based on the finite element formulation requires matrix inversion, which is computationally expensive if the matrix size is large. If the geometric domain is large and complex, the sizes of matrices **K** and **M** can also be very large. Therefore, improving the computational efficiency is important for PBCS for high-resolution measurement. Here, a domain decomposition method is proposed to improve the efficiency in PBCS recovery. Since the global conduction matrix **K** and mass matrix **M** are obtained by assembling the local conduction and mass matrices element-by-element, the temperature value computed at a node is only affected by its neighbouring elements. The elements outside the local neighbourhood have little or no influence to this node. For example, when the temperature of a cube in Figure 8 is monitored, the thermal load conditions at the boundaries of the cube need to be recovered. If the nodal temperature values on the four side faces can be measured and the boundary conditions are applied on these faces, only the shaded region with the correlation distance of *c* will affect the load vector applied to the side faces. Therefore, the load vector can be recovered more efficiently without incorporating other portion of the model. With this strategy, the size of conduction matrix **K** and mass matrix **M** can be significantly reduced during recovery. The smaller the correlation distance *c* is, more efficient the recovery process is. The comparison of PBCS recovery accuracy with different values of *c* will be shown in Section 3.4.3.

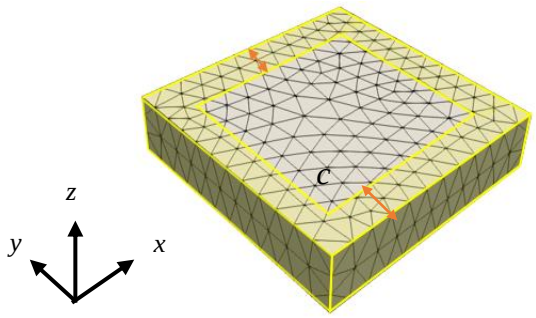


Figure 8. An illustration of domain decomposition

3.4.2 Physical experiments

Two examples are used to demonstrate the new PBCS method to monitor temperatures in the FFF process. The first example is a 45 mm×45 mm×45 mm cube. The second one is a gear with the root diameter 44 mm, tip diameter 51.2 mm, and the height 8 mm. They are printed with a Hyrel3D printer and the temperature field is measured with a Seek thermal infrared camera. The camera is fixed above the printer to have a top view of printed part. The Seek thermal infrared camera is calibrated with the FLIR T300 thermal camera by measuring the cooling process of the extruder from 224 °C to 80 °C. With the measurements from the FLIR T300 thermal camera as the reference, it is seen that the difference between the Seek thermal infrared camera and FLIR T300 is ±2.5 °C within the range using the root sum squared regression method. Printer settings are shown in Table 2. To evaluate the efficiency and accuracy of PBCS, the thermal camera measurements are used as the reference. A small subset of the measurement data from the camera are taken as if they were obtained by other single-point or low-resolution measurement instruments. These readings are used as the input of PBCS. This sampling approach is to remove the influence of systematic errors in instruments. The errors to be analysed and compared are from the PBCS method itself. The PBCS reconstructed temperature fields are compared with the original thermal images.

In this scenario, the printer was paused when the printed cube reached the size of 45 mm×45 mm×5 mm. The thermal camera measured the temperature distribution on the top surface as the part cools down, as shown in Figure 9. The time interval between two consecutive measurements was 1 s. The temperature distributions after the image registration by affine transformations are shown in Figure 10, where the value of each pixel in the grayscale images is linearly mapped to the actual temperature reading. The cooling process of the gear was also tested. The printer was paused when the part has the height of 4 mm. The temperature distributions of the gear after the similar image registration are shown in Figure 11. The PBCS method to monitor the cooling process will be demonstrated in Section 3.4.3.

Table 2. Printer settings

Parameters	
Fill density (%)	50
Layer height (mm)	0.3
Filament diameter (mm)	0.75
Bed temperature (°C)	80
Extruder temperature (°C)	235

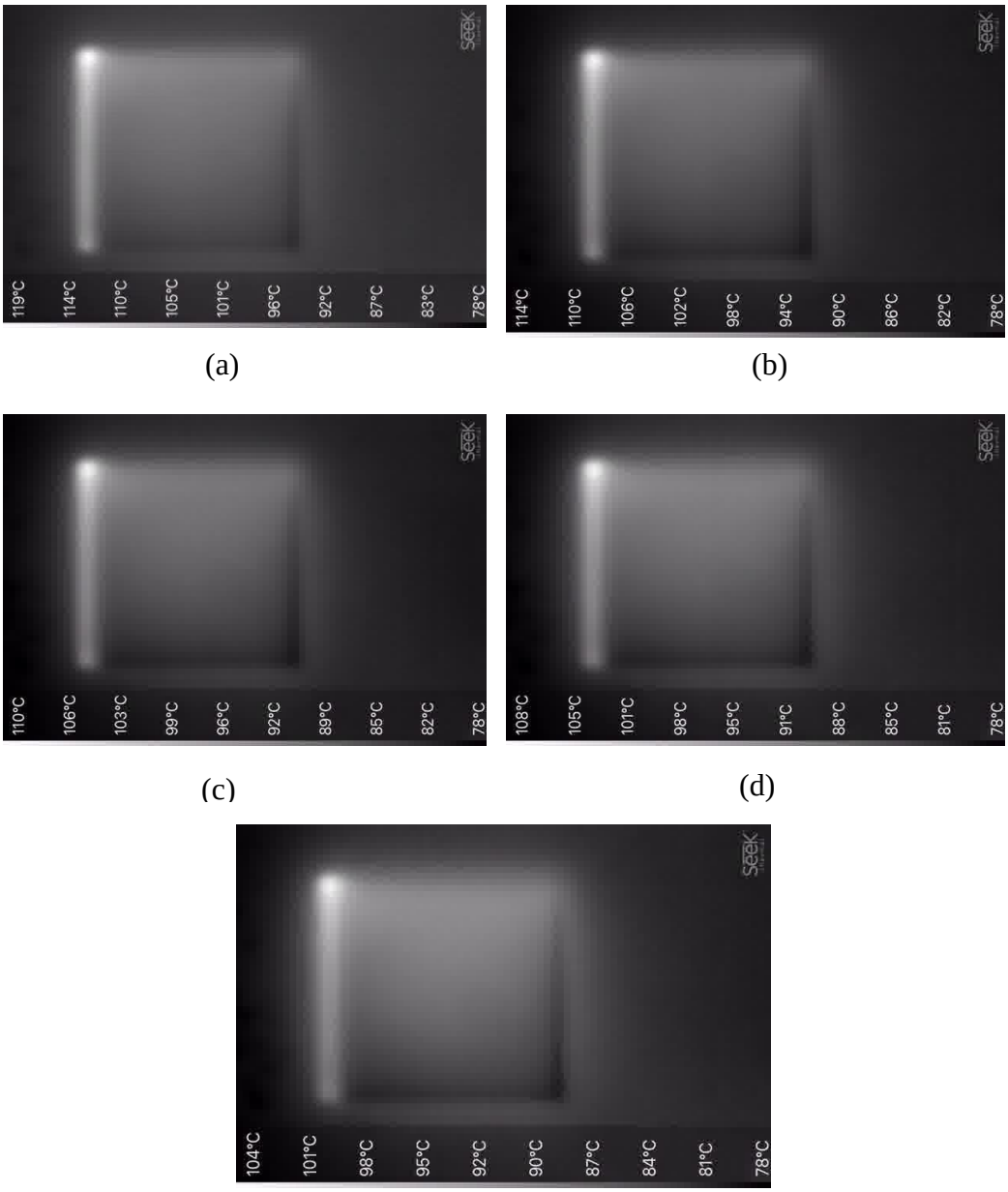
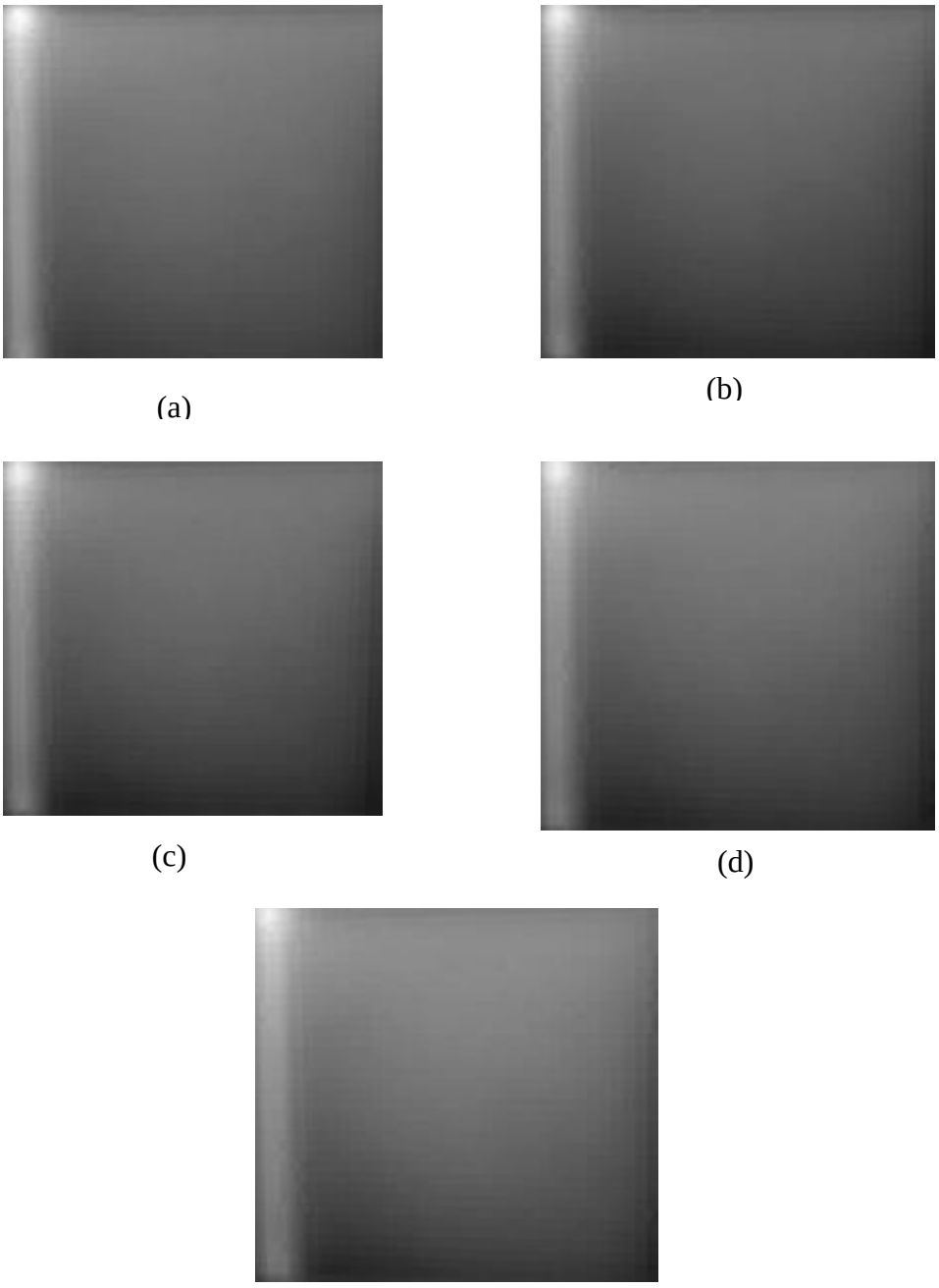


Figure 9. The full thermal images of the top surface for the cooling process of printed cube at (a) 0 s, (b) 1 s, (c) 2 s, (d) 3 s, and (e) 4 s.



(a) (b)

(c) (d)

(e)

Figure 10. The temperatures of top surface after image registration from full images in Figure 9 for time steps (a) 0 s, (b) 1 s, (c) 2 s, (d) 3 s, and (e) 4 s.

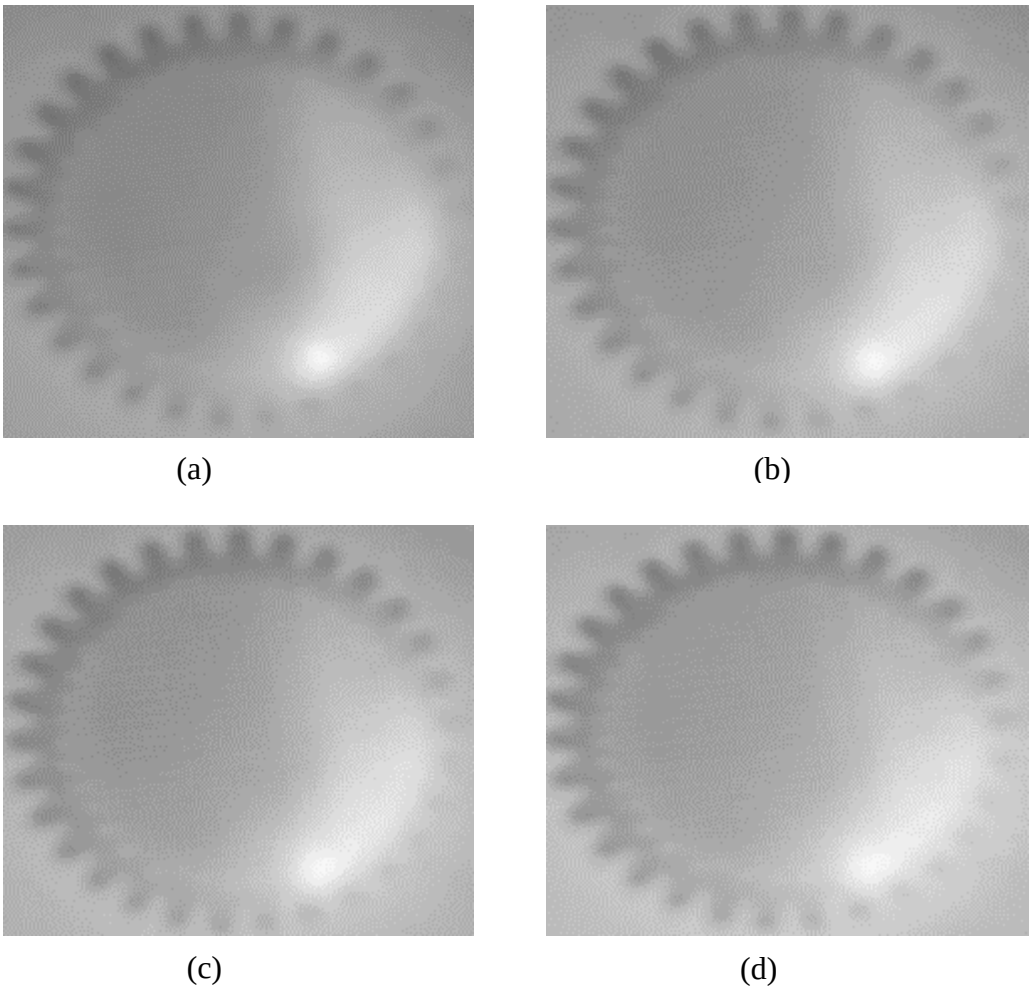


Figure 11. The measured 2D temperature distributions of top surface during the cooling process of the printed gear after image registration at (a) 0 s, (b) 1 s, (c) 2 s and (d) 3 s.

3.4.3 PBCS to Monitor Cooling Process

The framework of PBCS is implemented in Python language. The heat transfer modelling with finite element formulation is developed based on FEniCS [111]. The application of PBCS to monitor the cooling process is used to demonstrate the efficiency improvement when the domain decomposition method in Section 3.4.1 is applied, in comparison with original PBCS. The different values of correlation distance c are used to analyse its effect on the computational cost and accuracy for sensitivity studies.

3.4.3.1 Case 1: Monitor cooling process of the cube

The geometric model of the printed cube is shown in Figure 5, with three newly printed line segments attached on the top left of the part, which matches the case in the experiment. The geometry remains constant and the temperature changes along time. The corresponding mesh model in Figure 12 has the maximum mesh size of 2mm and a total of 2416 nodes.

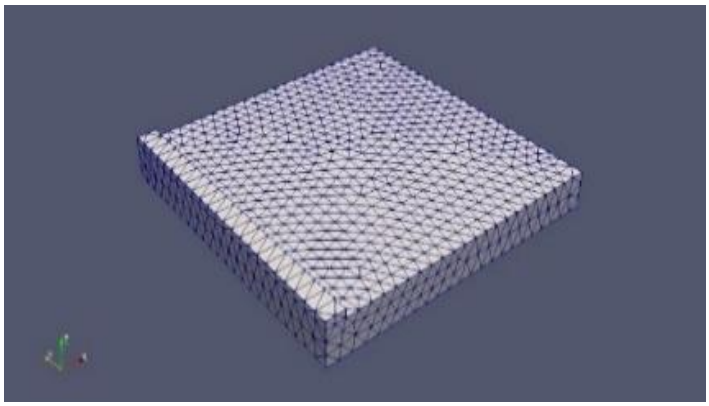


Figure 12. the mesh model.

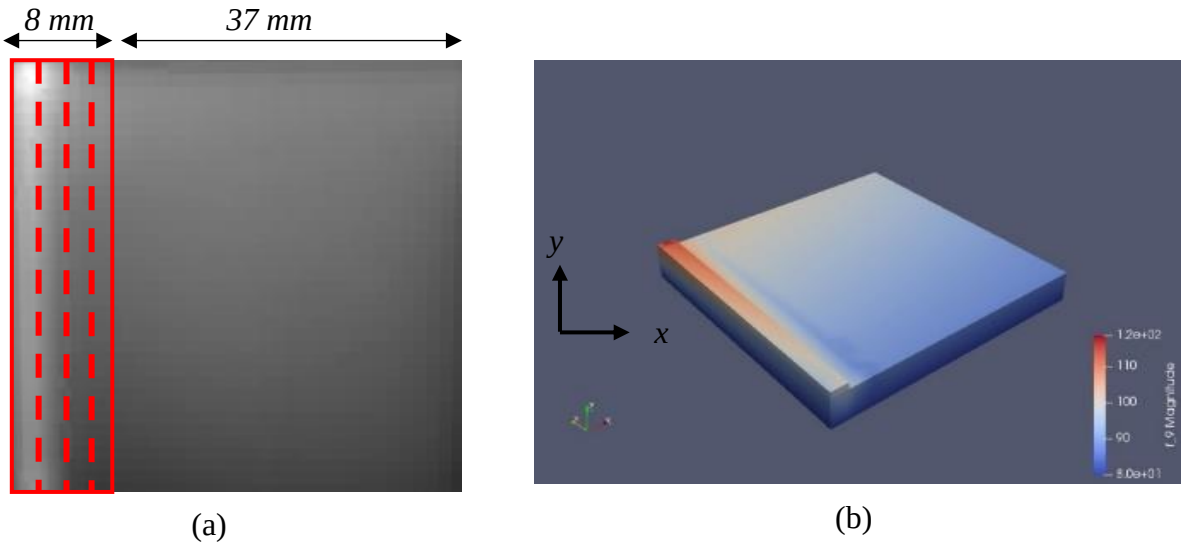


Figure 13. (a) 20 temperature readings are taken in high- and low-gradient regions; (b) initial temperature distribution at time step 0 s after interpolation.

The initial temperature field $\boldsymbol{T}_0$ is needed to recover load vector, as shown in Eq.(3.16). The initial temperatures at the top surface can be estimated by interpolating a few point-wise temperature readings taken from Figure 10 (a), assuming that only low-resolution single-point measurements are available instead of full thermal images. As shown in Figure 13 (a), eight temperature readings are taken along each of the top and bottom edges in the left region (F11 and a portion of F6) enclosed by the box. This region has high temperature gradients. The temperature distribution along *y* direction in this region is estimated by linearly interpolating the temperature readings on top and bottom edges. The temperature distribution on the right region, where temperature gradients are low, is predicted by bilinear interpolation with the temperature readings at the four corners. The initial temperatures on bottom face F5 are assumed to be the hotbed temperature 80 °C. The initial temperatures inside the part are estimated by linearly interpolating the top and bottom surface temperatures along *z* direction. The temperatures on side faces F1 to

F4 are similarly obtained from the interpolation based on the 20 measurements taken previously at the edges and corners in Figure 13(a). The interpolated initial temperature field is shown in Figure 13(b). During the monitoring process, another 20 temperature readings on the edges and corners at time 4s are used to recover and reconstruct temperature fields at time steps 1s, 2s, 3s, and 4s. The compression ratio is (2416×5)/(20×2)=302. The reconstructed temperature fields for the four time steps are shown in Figure 14.

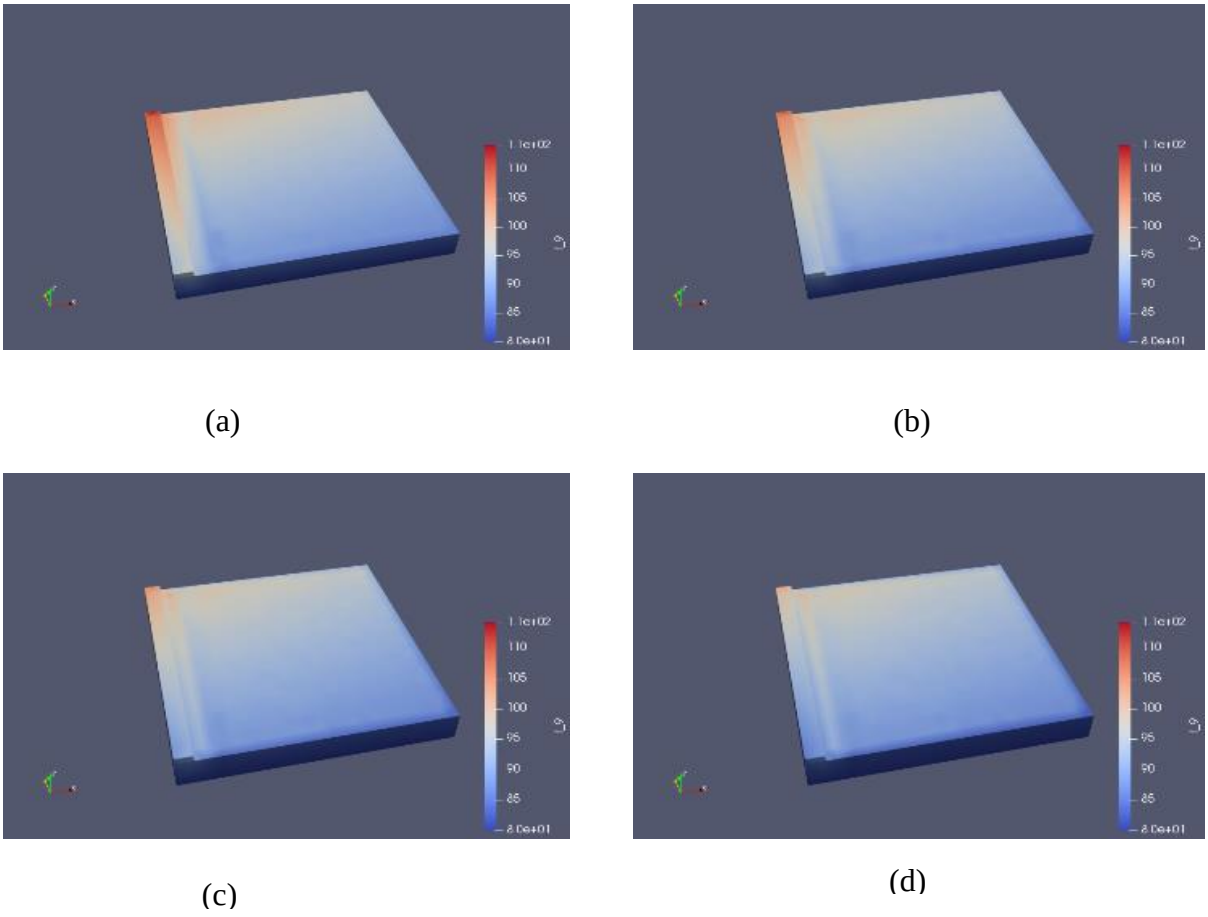


Figure 14. The reconstructed 3D temperature fields based on the PBCS low-resolution measurement scheme at time steps (a) 1 s, (b) 2 s, (c) 3 s, and (d) 4 s for the cube.

By comparing the 2D temperature distribution on the top surface interpolated from the reconstructed nodal temperatures with the original thermal camera images, the PBCS reconstruction errors can be estimated. The pixel-by-pixel PBCS reconstruction errors for the top surface are shown in Figure 15. The average errors for the four steps are 2.73%, 3.31%, 2.89%, and 2.89% respectively, and the corresponding standard deviations are 1.03%, 1.12%, 1.18%, and 1.27%. The computational time for recovery is 14 s when the full size of matrices **K** and **M** (2416×2416) is used. The reconstructed results in Figure 14 are used as the baseline to compare the performance of the improved PBCS with domain decomposition where different values of correlation thickness c are used.

The correlation distances of 0 mm, 4 mm, and 8 mm are used in sensitivity studies of the improved PBCS with domain decomposition. The nodes in shaded region shown in Figure 8 are used to form global matrices **K** and **M**. As the value of c increases, the size of the matrices grows. The computational time for 0 mm, 4 mm, and 8 mm are 4.27 s, 5.44 s, and 7.77 s, with matrix sizes of 307×307, 912×912 and 1504×1504 respectively. The errors of reconstructed temperatures at 4s are compared in Figure 16. The average relative differences between the original PBCS with full matrices and the improved PBCS with domain decomposition are 4.69%, 0.27% and 0.0015%, and the corresponding standard deviations 11.57 %, 1.13% and 0.0056 %. It is seen that the reconstruction failed when the correlation distance of 0 mm is used. The results with the correlation distance of 8 mm are very close to the original full-matrix PBCS, yet with only almost half of the computational cost.

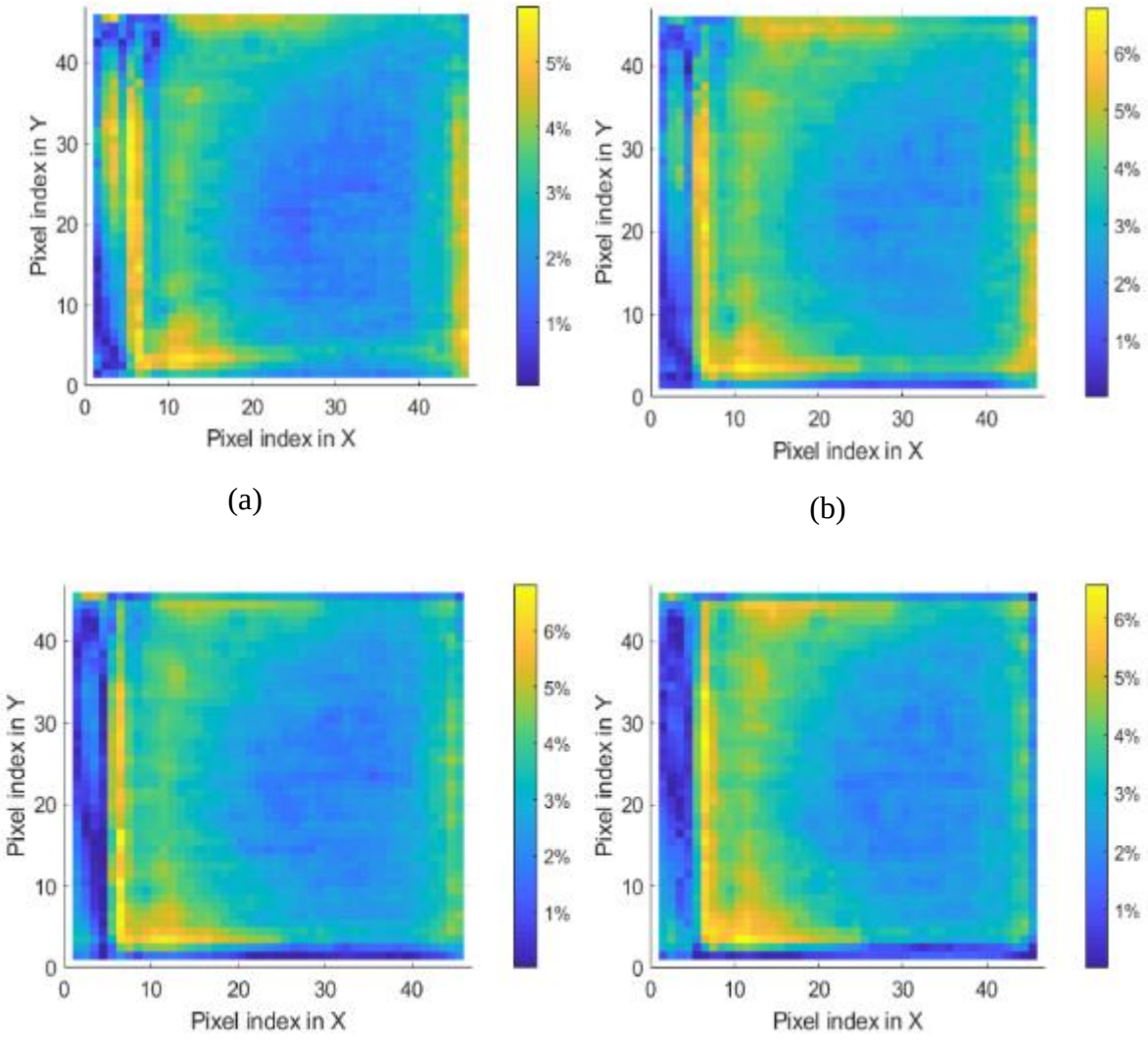


Figure 15. Pixel-by-pixel reconstruction errors for the top surface with full-matrix PBCS at time steps (a) 1 s, (b) 2 s, (c) 3 s, and (d) 4 s.

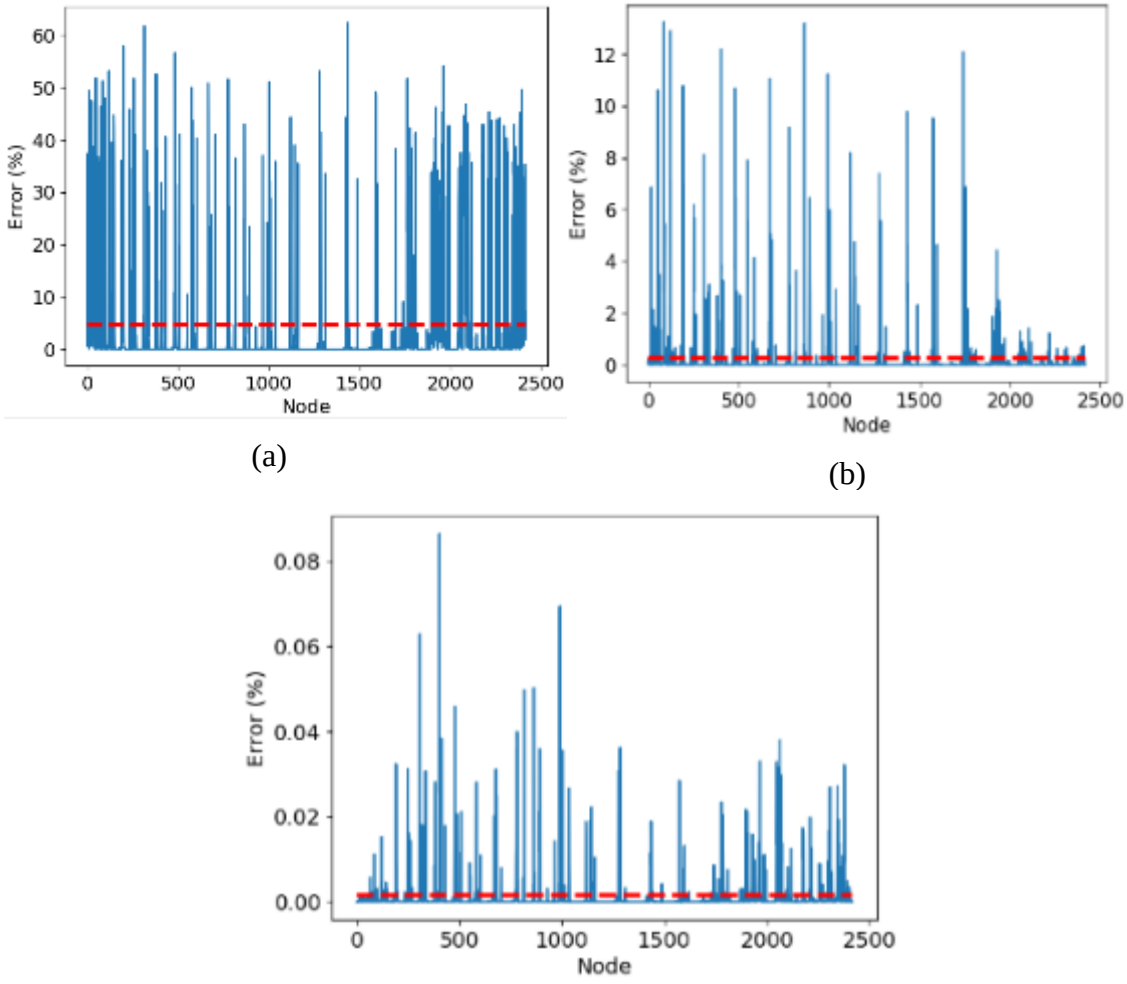


Figure 16. The reconstructed nodal temperature differences between the original full-matrix PBCS and the domain decomposition method at 4 s using correlation distances of (a) 0 mm, (b) 4 mm, and (c) 8 mm for the cube.

The major source of errors in PBCS is the physical model, where assumptions and simplifications are made during the modelling process. These errors can be treated as systematic errors and are reducible by introducing more accurate models, reducing geometry inaccuracy and environmental fluctuation, or by introducing multi-physics models. Instead of refining the physics-based model, a data-driven approach is taken here to quantify the systematic error. A Gaussian process regression (GPR) model is applied to predict the errors and compensate the errors. GPR is used to model the difference between the predicted temperature distribution by PBCS and the measured one by full thermal imaging, based on the sampling in both spatial and temporal domains. The GPR model is then used to predict the systematic error for any particular location and time, which can be applied for the error compensation purpose. For the GPR model, the first order polynomial basis function and the exponential covariance function are used. To evaluate the GPR model, the coordinates of each pixel in *x* and *y* axes, and the time step of the image are used as the input, and the output is the difference between reconstructed temperature readings on the top surface extracted from Figure 14 and measurements in Figure 10 (b-e). Each thermal image in Figure 10 (b-e) has the size of 46×46 pixels. Among a total of 2116 pixels, 600 sampling points at each of the time steps except time step 3s are randomly selected to construct the 3D GPR model. The PBCS reconstruction errors after error compensation at all pixel positions and all four time steps are shown in Figure 17. The average errors for the four steps are 0.13%, 0.14%, 0.26 %, and 0.14% respectively, and the corresponding standard deviations of errors are 0.19%, 0.21%, 0.29%, and 0.27%. By comparing average errors and standard deviations of errors before error compensation in Figure 15, it is seen that error predictions for all four time steps are significantly reduced. The extent of error reduction at time step 3s is not as much as at other three time steps. This is because error information at time step 3s is not considered in the GPR model construction,

and there are not enough samples in the time series. If more time steps are considered and more samples for each time step are taken, the error prediction can be more accurate. In this example, it takes about 7.91 seconds of computational time to build the GPR model and 23.28 seconds to predict errors at all pixels for all four time steps.

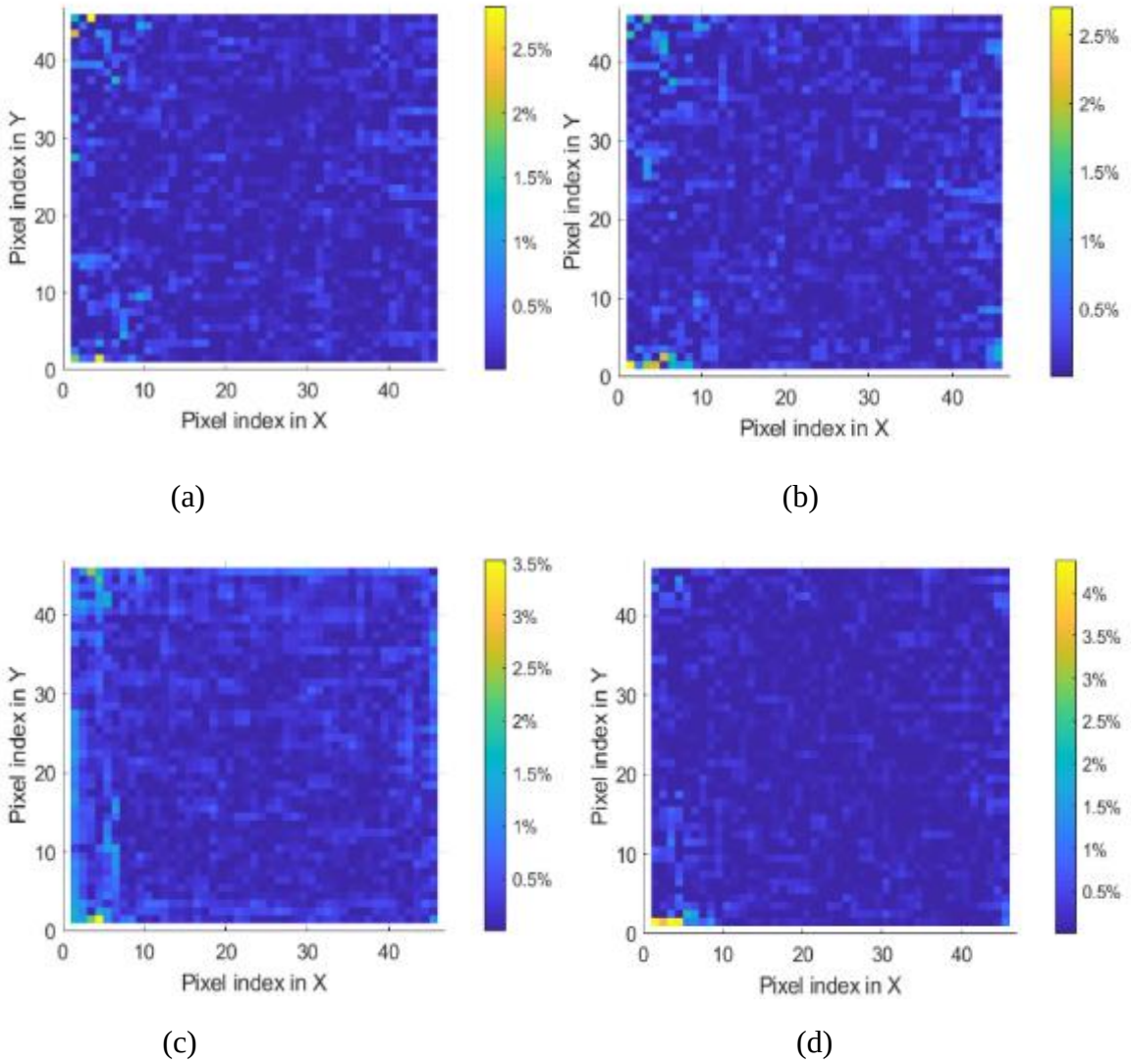
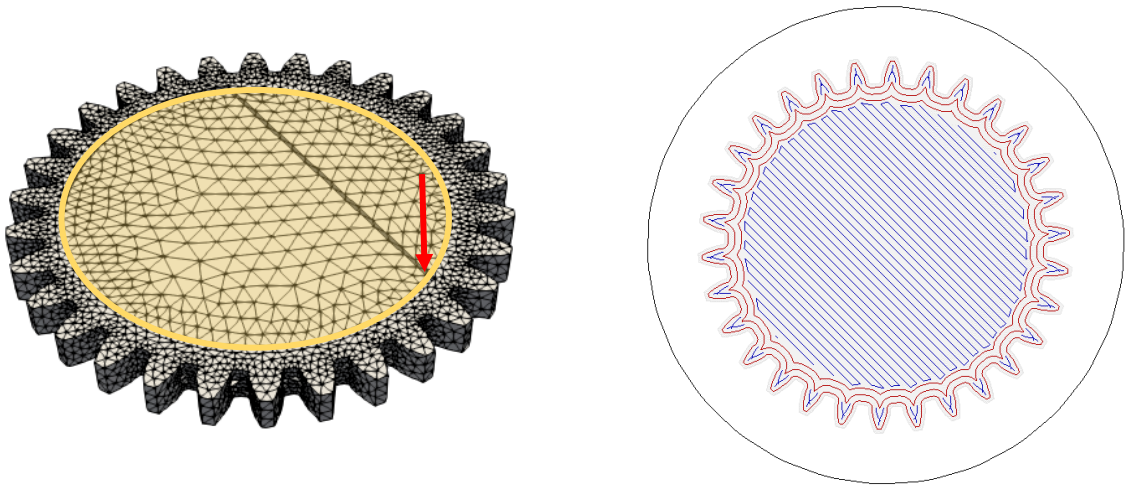


Figure 17. Pixel-by-pixel reconstruction errors for the top surface after error compensation at time steps (a) 1 s, (b) 2 s, (c) 3 s, and (d) 4 s.

3.4.3.2 Case 2: Monitor cooling process of the gear

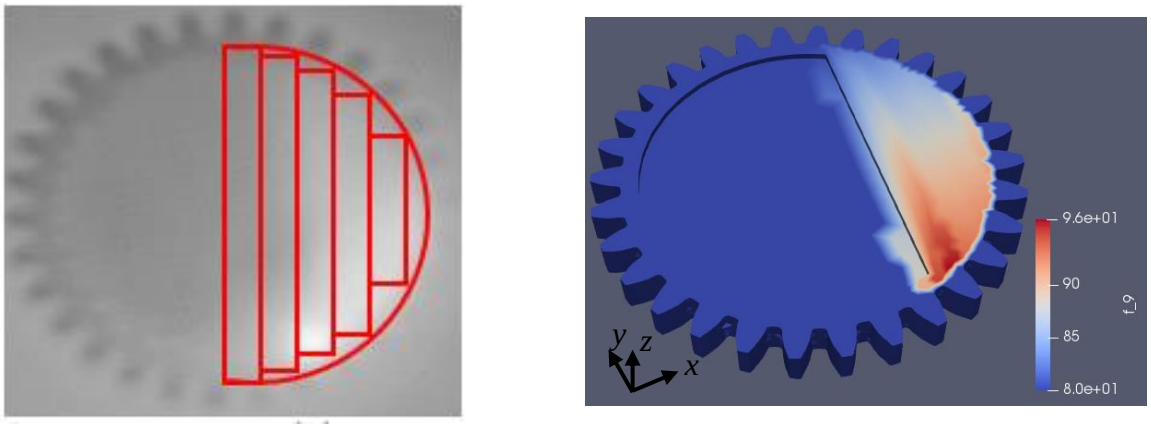
The example of gear has more complex geometry. The mesh model of printed gear is shown in Figure 18(a) with a total of 5980 nodes. The printer was paused at the location indicated by the arrow. The printing path of the top layer is shown in Figure 18 (b). The contours of the gear teeth are printed first. Then the inside region is filled. Since the side faces of the gear only have a small contact area with air, the effect of convection cooling is negligible in the physical model. Only the convection from the shaded region of the top face in Figure 18 (a), or the main body of the gear, is considered.



(a) (b)

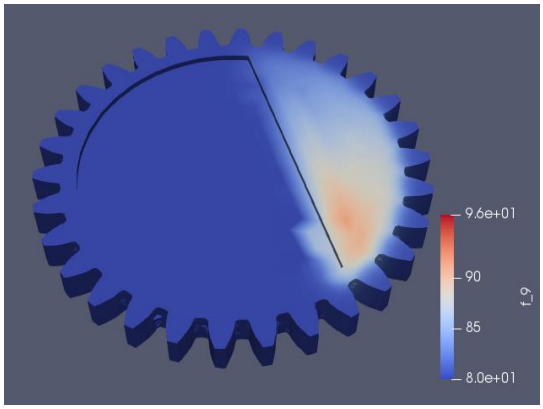
Figure 18. (a) The mesh model of printed gear and (b) the printing path of the top layer

The initial temperature distribution is estimated from a few temperature readings of the top surface as in Figure 11(a). As illustrated in Figure 19(a), for the high temperature gradient region on the right where new line segments are printed, the temperature distribution of the top surface is estimated by linearly interpolating between 8 temperature readings at the upper half contour and 8 at the lower half contour. Other regions of the top surface have cooled down for a while and are assumed to have the constant temperature with only one measurement. The bottom surface is assumed to have the same temperature as the hotbed. The temperatures inside the gear are estimated by interpolating between the top and bottom surface temperatures. The initial temperature distribution after interpolation is shown in Figure 19(b). With 17 measurements at time 3s, temperature distributions at 1 s, 2 s and 3 s are reconstructed by PBCS and shown in Figure 20.

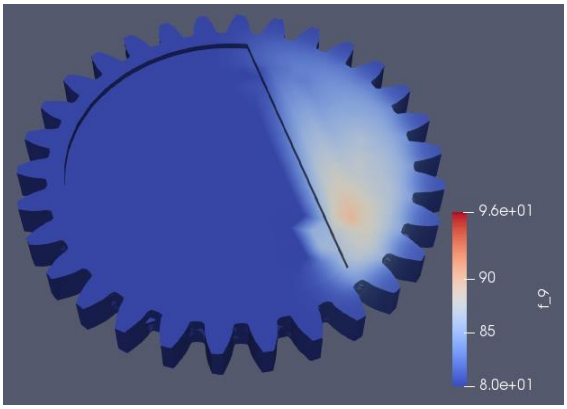


(a) (b)

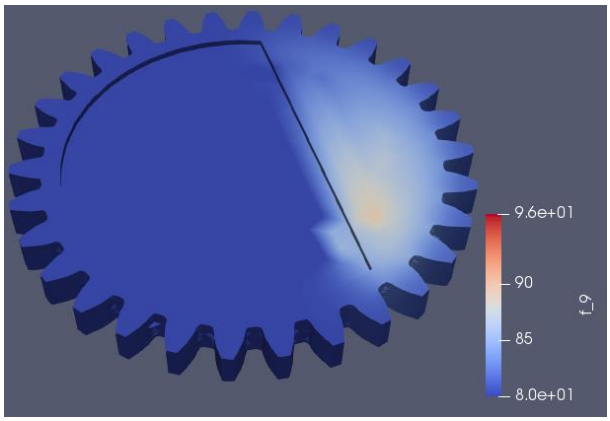
Figure 19. (a) 17 measurements to interpolate the initial temperature of the gear (b) initial 3D temperature distribution from the interpolation



(a)



(b)

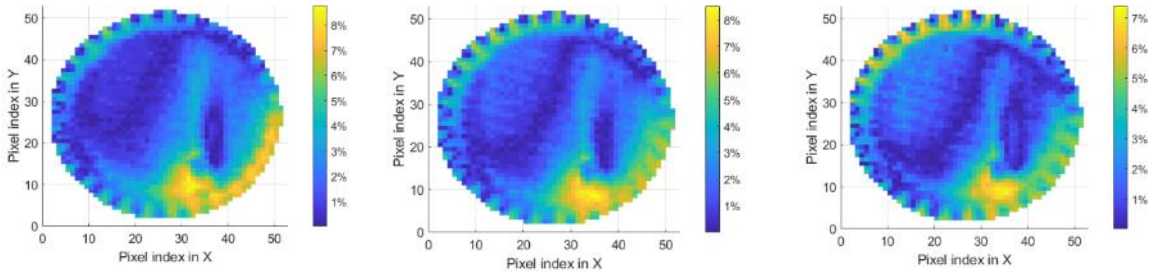


(c)

Figure 20. The reconstructed 3d temperature fields based on the PBCS low-fidelity measurement scheme at time steps (a) 1 s, (b) 2 s and (c) 3 s for the gear.

With 16 measurements in the high temperature gradient region and 1 measurement for the other regions, the compression ratio is (5980×4)/(17×2)=703.5. The PBCS reconstruction errors are shown in Figure 21. The average errors indicated by the dash lines for the three time steps are 2.52%, 2.46%, and 2.20 % respectively, and the corresponding standard deviations are 2.07%, 1.89% and 1.69%. The computational time for recovery is 69 s when the full size of matrices **K** and **M** (5980×5980) is used.

The domain decomposition method is then applied, as illustrated in Figure 22(a). The correlation distance, *c*, is the radius of the shaded cylindrical domain. Only nodes in the shaded cylindrical neighbourhood are used to assemble the model matrices. When the correlation distance is 20.3 mm, the size of matrices **K** and **M** is 2376×2376. The computational time for recovery is 9.5 s, which is only 1/7 of the computational time with the full-matrix PBCS. The nodal temperature differences between the domain decomposition method and the original PBCS for the three time steps are shown in Figure 22(b)-(d). The average relative differences are 0.007%, 0.013%, and 0.021% respectively, and the corresponding standard deviations are 0.045%, 0.062% and 0.073%.



(a) (b) (c)

Figure 21. Pixel-by-pixel reconstruction errors for the top surface at time steps (a) 1 s, (b) 2 s and (c) 3 s for the gear.

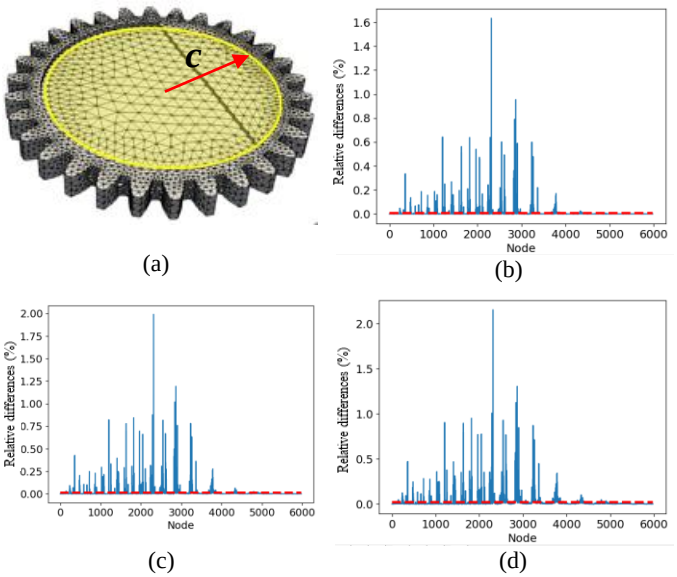
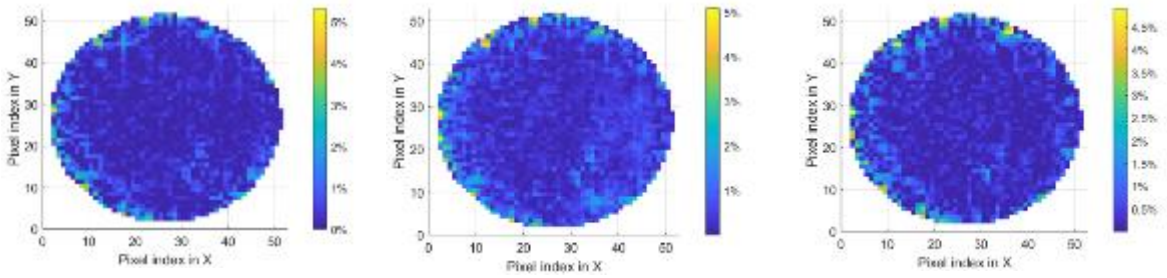


Figure 22. (a) The domain decomposition in the gear model with cylindrical neighborhood; the nodal temperature differences between the domain decomposition method and full-matrix PBCS at time steps (b) 1 s, (c) 2 s, and (d) 3 s.

GPR is used to model the difference between reconstructed temperature readings on the top surface of the gear in Figure 20 and measurements in Figure 11. Each thermal image in Figure 11 has the size of 52×52 pixels. Among a total of 2704 pixels, 700 sampling points at time steps of 1s and 3s are randomly selected to construct the 3D GPR model. The PBCS reconstruction errors after error compensation at all pixel positions and all three time steps are shown in Figure 23. The average errors for the three steps are 0.42%, 0.58%, and 0.44% respectively, and the corresponding standard deviations of errors are 0.68%, 0.66%, and 0.69%. Compared to average errors and standard deviations in Figure 21, the reconstruction errors after error compensation are significantly reduced. It takes about 10.51 seconds of computational time to build the GPR model and 22.14 seconds to predict errors at all pixels for three time steps.



(a) (b) (c)

Figure 23. Pixel-by-pixel reconstruction errors for the top surface at time steps after error compensation at (a) 1 s, (b) 2 s and (c) 3 s for the gear.

3.5 Demonstration of PBCS for real-time model update

3.5.1 Physical experiments

The PBCS model for real-time printing process is more complex, where the modelling domain also evolves along time. In this scenario, the thermal camera was used to measure the temperature distribution on the top surface of the cube in real time during the printing process. Seven of the measurements are shown in Figure 24. The measured temperature distributions after image registration are shown in Figure 25.

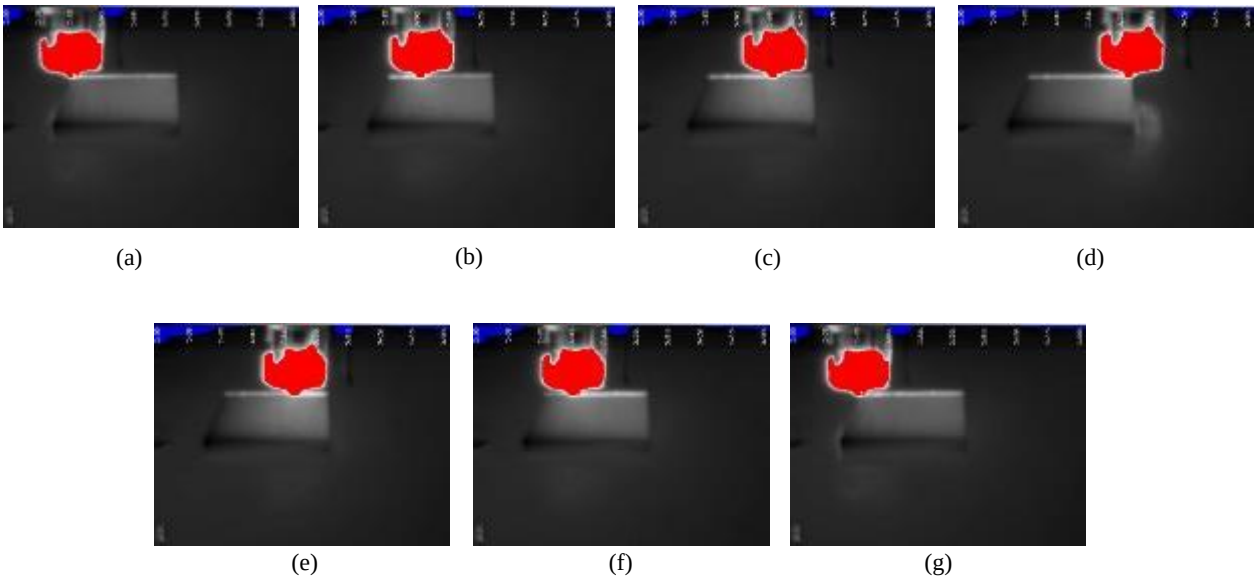


Figure 24. The full thermal images of the top surfaces for the real-time monitoring of the printed cube at (a) 0 s, (b) 1/3 s, (c) 2/3 s, (d) 1 s, (e) 4/3 s, (f) 5/3 s, and (g) 2 s.

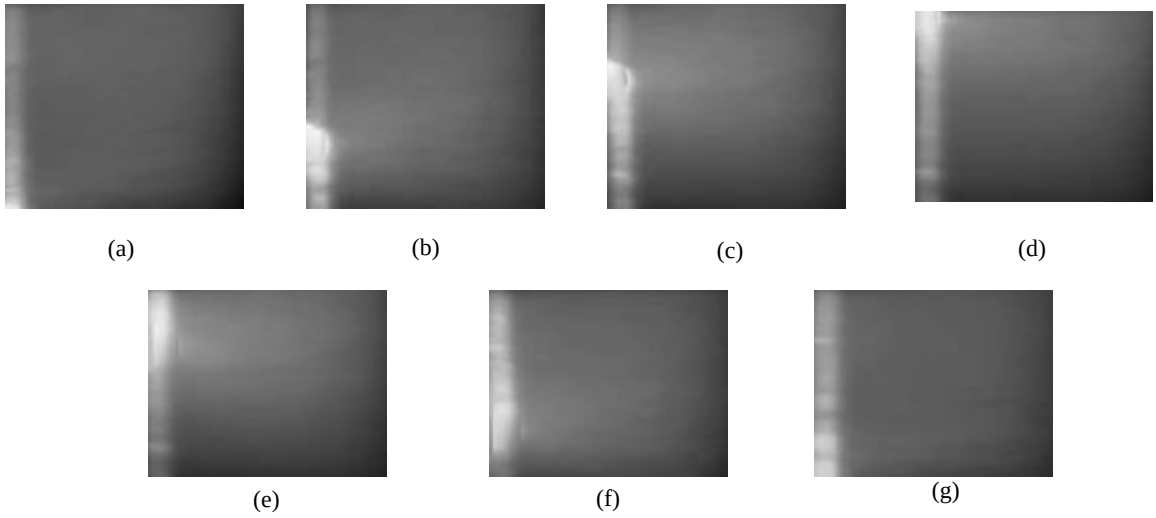


Figure 25. Temperature meausurements of the top surface from Figure 24 after image registration at (a) 0 s, (b) 1/3 s, (c) 2/3 s, (d) 1 s, (e) 4/3 s, (f) 5/3 s, and (g) 2 s.

3.5.2 PBCS for Real-Time Model Update

The real-time monitoring of the temperature distribution with PBCS requires the consideration of geometry changes in the printing process. New material is continuously deposited on the top

surface of the part along time. The method of birth-and-death element approach is used to update matrices **K** and **M** at each time step in order to perform the real-time model update. The complete matrices $\mathbf{K}_f$ and $\mathbf{M}_f$ at the final time step are first generated. At the initial time step, all elements are deactivated. The conduction matrix $\mathbf{K}_0$ at the initial time is generated by multiplying $\mathbf{K}_f$ with a factor of 10^{-6}. The mass matrix $\mathbf{M}_0$ at the initial time is also zeroed out. After a new segment is printed, the corresponding elements in the new segment are activated by returning the conduction and mass matrices gradually to their original values.

The domain decomposition method is also used to improve the efficiency. As illustrated in Figure 26, only nodes within the shaded region on the left, with the correlation distance of 30mm, are used to form matrices **K** and **M**. In this region, temperature gradients are higher because of the heat from the extruder. With the smaller size of matrices **K** and **M**, associated cost of matrix computation in the recovery process can be reduced. With a few measured temperature readings, the heat flux on the top surface within the shaded region can be recovered. The heat flux on the top surface outside of the shaded region is assumed to have the same value as the heat flux along the domain boundary marked as the dash line in Figure 26. The printing process for a time period of 2s is monitored. The time step in the physical model is 1/9 s. The initial temperature distribution is obtained by interpolating a few measured temperature readings of the thermal image in Figure 25(a). As shown in Figure 27, the thermal image at 0s is divided to two separate regions. The left region enclosed by the solid line includes newly printed line segments with high temperature gradients. The remaining region has lower temperature gradients. For the region with high temperature gradients, the initial 2D temperature distribution in this region is obtained by interpolating 18 measured temperature readings marked with the circles in Figure 27. For the region with low temperature gradients, 24 measured temperature readings marked with the small

squares are used to approximate the initial 2D temperature distribution in this region by interpolation. After the initial temperature distribution on the top surface is interpreted, the complete 3D temperature distribution is also estimated by interpolating the temperatures on top and bottom faces. The temperature on the bottom face is assumed to be constant, which is the temperature of the hotbed. Therefore, the initial temperature distribution is estimated with 42 measurements.

To recover the heat load vector, a few temperature readings at 1/3s, 2/3s, 1s, 4/3s, 5/3s and 2s in Figure 25(b)-(g) need to be measured. Temperature readings at different time steps are extracted similarly as the initial temperature in Figure 27, but only 30 measurements within the correlation distance of 30mm on the left are used to recover the heat load vector. More temperature readings measured on the top surface can improve the accuracy of reconstruction. With a few temperature readings on the top surface measured at 1/3s, 2/3s, 1s, 4/3s, 5/3s and 2s, the intermediate temperature distributions can be reconstructed as shown in Figure 28, where images of every 2/9 s are shown. The average reconstruction errors at 1/3s, 2/3s, 1s, 4/3s, 5/3s and 2s shown in Figure 29 are 1.56%, 1.8%, 1.86%, 1.76%, 1.94% and 1.81% respectively, and the corresponding standard deviations are 1.97%, 1.93%, 1.81, 1.79%, 2.45% and 2.25%. It is seen in Figure 29 that most reconstruction errors come from the region of newly printed line segments, because the approximation of the initial temperatures in this region with interpolation can introduce large errors. Reconstruction errors can be reduced if more measurements in the high temperature gradient region are available.

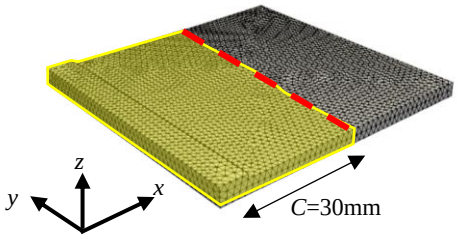


Figure 26. Domain decomposition for real-time monitoring

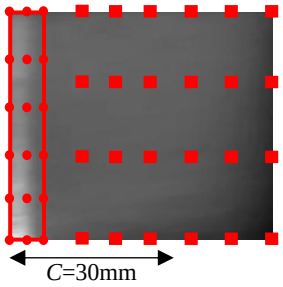


Figure 27. Approximation of the initial temperature distribution at 0s for the real-time monitoring

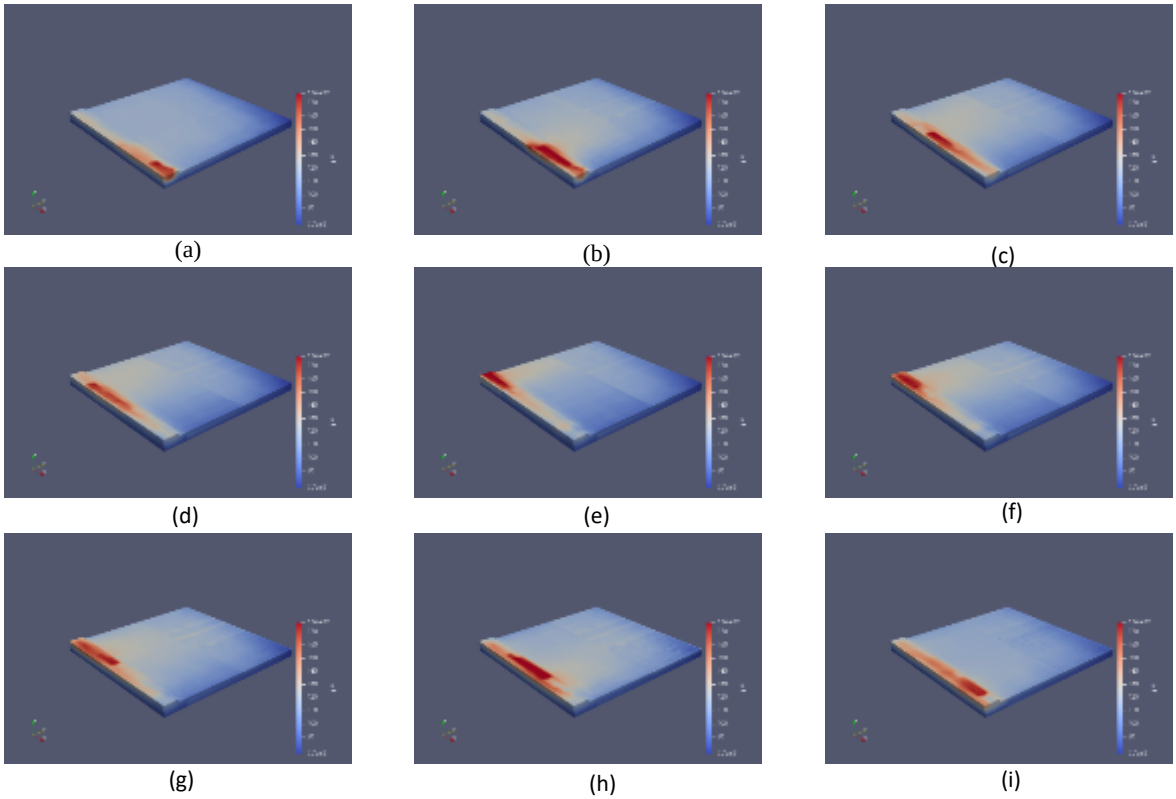


Figure 28. Reconstructed 3D temperature distributions based on the PBCS real-time scheme, where images are shown for every 2/9 s.

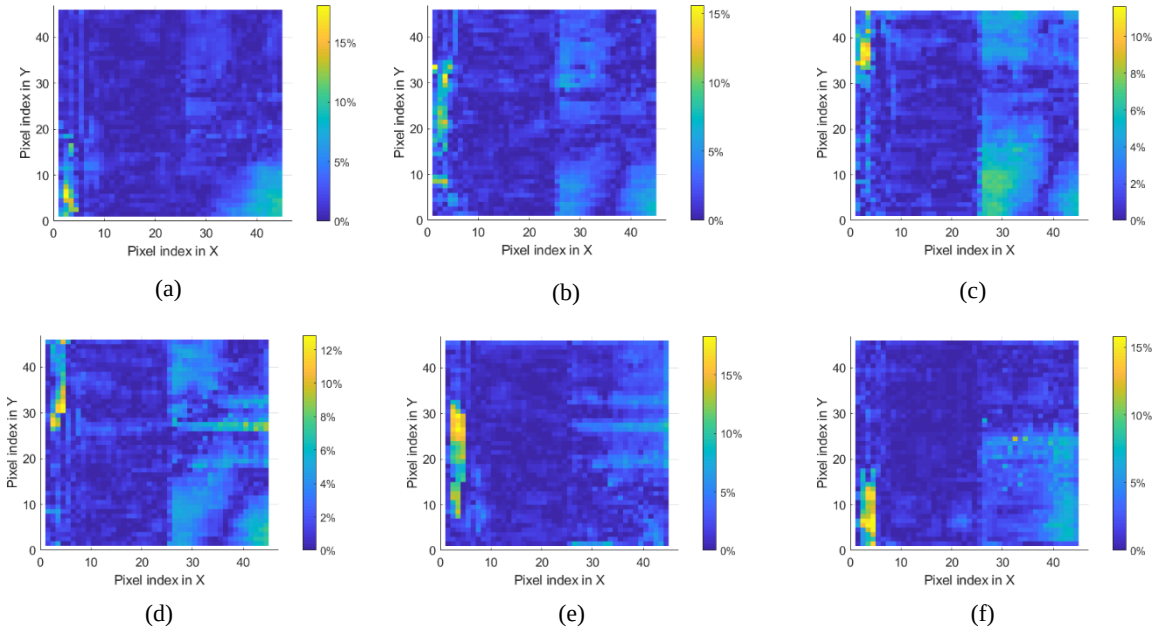


Figure 29. Pixel-by-pixel PBCS errors of the top surface at (a) 1/3 s, (b) 2/3 s, (c) 1 s, (d) 4/3 s, (e) 5/3 s, and (f) 2 s.

3.6 Conclusion

The PBCS approach to efficiently monitor the temperature distribution of the FFF process is proposed, where temperature information can be obtained from limited sensor data by incorporating the heat transfer model and numerical methods in CS. Compared to the traditional CS, the PBCS approach can significantly improve the compression ratio and use low-cost sensors to replace high-cost ones. In experiments, it is demonstrated that only a few measurements for temperatures in the 3D printing domain, such as the side faces and a few readings on top faces in the printed part, are necessary to reconstruct the complete 3D temperature distribution. In the real-time monitoring process, the PBCS model has different conductivity and mass matrices at each time step. The birth and death element approach is applied in modeling. The prior knowledge of the thermal model is also used as the constraint in a constrained OMP algorithm to efficiently recover the heat load vector. A domain decomposition method is introduced to reduce the sizes of

conduction and mass matrices in the PBCS model, thus improving the efficiency of the recovery process. GPR is also used to predict and compensate PBCS errors for more accurate reconstruction. With less amount of data collected and fewer sensors deployed, the proposed PBCS shows its advantages over traditional CS for process monitoring.

CHAPTER 4. Fluid Velocity Field Monitoring

4.1 Introduction

To monitor the fluid velocity field efficiently, a PBCS approach is proposed based on fluid flow models [112]. There has been some research on solving inverse problems in CFD modelling, which is reviewed in Section 2.3. The objective of those approaches was to either optimize the input parameters or reconstruct the flow or pressure fields with the available measurements. In those approaches, the inverse problems were solved by traditional optimization algorithms such as the genetic algorithms and conjugate gradient algorithm. These algorithms suffer from the instability issue in solving these inverse problems. The computation can also be expensive when the computational domain is large and complex. Different from the above approaches, the proposed PBCS mechanism takes advantages of data sparsity in formulating and solving the inverse problem, and as a result the numerical stability and robustness are significantly improved.

In the remainder of this Chapter, a laminar flow model represented as the Navier-Stokes equations is described in Section 4.2, PBCS formulations for the steady state and the transient problems are also implemented. Based on the Reynolds Averaged Navier-Stokes equations, the PBCS framework is developed to monitor the turbulent velocity field in Section 4.3. In Section 4.4, the PBCS framework is demonstrated with the steady-state backward-facing step flow, and the 2D reconstruction result is compared with experimental measurements. The advantages of the proposed PBCS framework and its challenges in physical applications are discussed in Section 4.5.

4.2 PBCS formulation for the laminar velocity field

4.2.1 Laminar flow model

To reconstruct the laminar velocity field, the objective function in Eq. (1.1) can be revised to minimize the difference between measured and reconstructed velocities. The constraint in Eq. (1.2) can be represented by the Navier-Stokes equations. The Navier-Stokes equations can be represented as the conservation of mass in a form of

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{4.1}$$

and the conservation of momentum as

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i}\delta_{ij} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] - \rho f_i \tag{4.2}$$

where index *i* indicates the *x*-, *y*-, or *z*-component of the vector field, u_i donates the velocity, p is the pressure, ρ is the density, μ is the dynamic viscosity of the fluid, f_i is the volumetric force and δ_{ij} is given as

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{4.3}$$

The Dirichlet inflow boundary condition for u_i is

$$u_i = g_i^{(1)} \tag{4.4}$$

where $g_i^{(1)}$ is the normal inflow velocity. The Neumann inflow boundary condition for u_i is

$$-p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = -g_i^{(2)} \tag{4.5}$$

where $g_i^{(2)}$ is the pressure applied at the inflow boundary. More generally, the inflow boundary condition can be expressed as

$$\left[-p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + hu_i = (h-1)g_i \tag{4.6}$$

When $h = 0$, Eq.(4.6) is the Neumann condition. When $h \to \infty$, Eq.(4.6) approximates the Dirichlet condition. In our implementation, $h = 10^8$ is used for the Dirichlet inflow boundary

condition.

4.2.2 Finite-element discretization of laminar flow model

After introducing test functions q and v_i, the weak formulation of Eqs. (4.1) and (4.2) over an element domain Ω^{e} can be expressed as

$$\int_{\Omega^{\mathrm{e}}} q \frac{\partial u_i}{\partial x_i}\, d\mathrm{x} = 0 \tag{4.7}$$

$$\int_{\Omega^{\mathrm{e}}} \left\{ \rho \left(v_i \frac{\partial u_i}{\partial t} + v_i u_j \frac{\partial u_i}{\partial x_j} \right) + \frac{\partial v_i}{\partial x_j} \left[-p\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] - \rho v_i f_i \right\} dx$$

$$- \int_{\partial\Omega_1} v_i (h-1) g_i ds + \int_{\partial\Omega_1} h v_i u_j ds - \int_{\partial\Omega_2} v_i \tau ds = 0 \tag{4.8}$$

respectively, where $\partial\Omega_1$ is the subdomain of the inflow boundary and $\partial\Omega_2$ is the subdomain of the solid walls. With shape functions ψ and ϕ, dependent variables u_i and p can be approximated as

$$u_i = \psi^{\,T} \boldsymbol{u}_i \tag{4.9}$$

$$p = \phi^T \boldsymbol{p} \tag{4.10}$$

By replacing v_i to ψ and q to ϕ. Eqs. (4.37) and (4.38) can be reformulated as

$$-\mathbf{Q}^T \boldsymbol{u} = \mathbf{0} \tag{4.11}$$

$$\mathbf{M_u} \dot{\boldsymbol{u}} + \mathbf{C}(\boldsymbol{u})\boldsymbol{u} + \mathbf{K_u}\boldsymbol{u} - \mathbf{Q}\boldsymbol{p} = \boldsymbol{F}_{\mathbf{u}} \tag{4.12}$$

or in a matrix form as

$$\begin{bmatrix} \mathbf{M_u} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{C}(\boldsymbol{u}) + \mathbf{K_u} & -\mathbf{Q} \\ -\mathbf{Q^T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_{\mathbf{u}} \\ \mathbf{0} \end{bmatrix} \tag{4.13}$$

where the coefficient matrices are defined as

$$\mathbf{M_u} = \int_{\Omega^{\mathrm{e}}} \rho\psi\ \psi^{\,T} dx$$

$$\mathbf{C}(\boldsymbol{u}) = \int_{\Omega^{\mathrm{e}}} \rho \boldsymbol{u} \psi\ \frac{\partial \psi^{\,T}}{\partial x_j} dx$$

$$\mathbf{K_u} = \int_{\Omega^{\mathrm{e}}} \mu \frac{\partial \psi}{\partial x_i} \frac{\partial \psi^{\,T}}{\partial x_j} dx + \int_{\partial\Omega_1^e} h\psi\, \psi^{\,\mathbf{T}} ds$$

$$\mathbf{Q} = \int_{\Omega^e} \frac{\partial \psi}{\partial x_i} \phi^T dx$$

$$\boldsymbol{F}_{\mathbf{u}} = \int_{\Omega^e} \rho \psi \, f_i \, dx + \int_{\partial \Omega_1^e} \psi \, (h-1) g_i ds + \int_{\partial \Omega_2^e} \psi \, \tau ds \tag{4.14}$$

4.2.3 PBCS formulation for laminar flow

Based on the finite-element discretization of the laminar flow model, the PBCS mechanism is established. It consists of two steps, which are the recovery of load vector and the reconstruction of velocity field. In the recovery step, the inverse problem is solved to find the load vector $\boldsymbol{F}_{\mathbf{u}}$ in Eq. (4.12) based on some limited measurements of velocities $\boldsymbol{u}^*$. The load vector $\boldsymbol{F}_{\mathbf{u}}$ needs to be recovered because it is a sparse vector in the physical model. Sparsity is important for the efficiency and accuracy of PBCS recovery, similar to traditional CS which requires the sparse representations of signals. The sparsity of the load vector $\boldsymbol{F}_{\mathbf{u}}$ can fulfill the requirement. It can be recovered very efficiently with only a few measurements and a properly designed basis matrix. In the reconstruction step, the complete velocity field is reconstructed by solving the forward problem based on the recovered load vector.

4.2.3.1 Steady state problem

For the steady state model, the time derivatives in Eq.(4.12) are ignored. Therefore, Eq. (4.12) can be simplified as

$$\mathbf{C}(\boldsymbol{u})\boldsymbol{u} + \mathbf{K}_{\mathbf{u}}\boldsymbol{u} - \mathbf{Q}\boldsymbol{p} = \boldsymbol{F}_{\mathbf{u}} \tag{4.15}$$

Eq. (4.15) can be further linearized and rearranged to

$$\boldsymbol{u}_k - \boldsymbol{\alpha}\mathbf{Q}\boldsymbol{p}_{k-1} = \boldsymbol{\alpha}\boldsymbol{F}_{\mathbf{u}} \tag{4.16}$$

where $\boldsymbol{\alpha} = (\mathbf{C}(\boldsymbol{u}_{k-1}) + \mathbf{K}_{\mathbf{u}})^{-1}$. Velocity $\boldsymbol{u}_{k-1}$ and pressure $\boldsymbol{p}_{k-1}$ come from the previous iteration. Before the reconstruction of the complete velocity field, the load vector $\boldsymbol{F}_{\mathbf{u}}$ needs to be

recovered from some measured velocities first. The measurement matrix $\boldsymbol{\Phi}$ is constructed to store the indices or locations of the limited measurements out of $\boldsymbol{u}_k$, which has the similar form as Eq.(3.12). The selected values from the model prediction are $\tilde{\boldsymbol{u}} = \boldsymbol{\Phi}\boldsymbol{u}_k$ where *M* out of *N* velocity values are taken. Multiplying $\boldsymbol{\Phi}$ to both sides Eq. (4.16) for the steady states, we have

$$\boldsymbol{\Phi}\boldsymbol{u}_k - \boldsymbol{\Phi}\boldsymbol{\alpha}\mathbf{Q}\boldsymbol{p}_{k-1} = \boldsymbol{\Phi}\boldsymbol{\alpha}\boldsymbol{F}_{\mathbf{u}} \tag{4.17}$$

In comparison to linear equations $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\gamma}$ in the traditional CS, coefficients $\boldsymbol{u}_k - \boldsymbol{\alpha}\mathbf{Q}\boldsymbol{p}_{k-1}$, $\boldsymbol{\alpha}$, and $\boldsymbol{F}_{\mathbf{u}}$ in Eq. (4.16) are corresponding to $\boldsymbol{y}$, $\boldsymbol{\Psi}$, and $\boldsymbol{\gamma}$ in traditional CS respectively. However, different from those in traditional CS, the coefficients in the PBCS are derived from the physical knowledge of fluid flows. Constructing the basis matrix $\boldsymbol{\alpha}$ from the physical model helps to improve the compression ratio for PBCS. The PBCS recovery is to find load vector $\boldsymbol{F}_{\mathbf{u}}$ such that the difference between the experimental measurements $\boldsymbol{u}^*$ and the selected predictions $\tilde{\boldsymbol{u}} = \boldsymbol{\Phi}\boldsymbol{u}_k$ from model is minimized as

$$\min\|\boldsymbol{u}^* - \boldsymbol{\Phi}\boldsymbol{u}_k\|_2 \tag{4.18}$$

subject to the constraints in Eq. (4.16) and

$$\|\boldsymbol{F}_{\mathbf{u}}\|_0 < s_l \tag{4.19}$$

Since it is assumed that there is no volumetric force, all non-zero values in the recovered load vector $\boldsymbol{F}_{\mathbf{u}}$ come from boundary conditions. Therefore, velocities near the boundaries can be measured to ensure the accuracy of load vector recovery. With more direct measurements, the recovery accuracy can be improved. However the computational cost is also increased.

4.2.3.2 Transient problem

For the transient mode, the general formulation in Eq. (4.12) can be rearranged as

$$\boldsymbol{u}_k - \boldsymbol{\alpha}(\mathbf{Q}\boldsymbol{p}_k + (\Delta t)^{-1}\mathbf{M}_{\mathbf{u}}\boldsymbol{u}_{k-1}) = \boldsymbol{\alpha}\boldsymbol{F}_{\mathbf{u}} \tag{4.20}$$

where $\boldsymbol{\alpha} = \left((\Delta t)^{-1}\mathbf{M_u} + \mathbf{K_u} + \mathbf{C}(\boldsymbol{u}_{k-1})\right)^{-1}$ and Δt is the time step. The pressure $\boldsymbol{p}_k$ can be approximated as $\boldsymbol{p}_k = \boldsymbol{p}_{k-1} + (\boldsymbol{p}_{k-1} - \boldsymbol{p}_{k-2})$. That is, $\boldsymbol{p}_k$ is approximated by extrapolating from the previous two time steps. The load vector $\boldsymbol{F_u}$ is recovered by solving Eqs. (4.18-4.20) with the constrained OMP algorithm in Table 1. The complete flow velocity field can be reconstructed based on the forward modelling.

4.3 PBCS formulation for the turbulent velocity field

4.3.1 Turbulent flow model

To reconstruct the turbulent velocity field, the constraint in Eq. (1.2) can be represented by the eddy viscosity models, which is based on the Reynolds Averaged Navier-Stokes (RANS) equations. In this approach, the mean flow is the problem of interest. The instantaneous fluid velocity and pressure fields can be expressed as the sum of means and fluctuating components, as

$$u_i = U_i + u_i' \tag{4.21}$$

$$p = P + p' \tag{4.22}$$

where U_i and P are the mean quantities of velocity and pressure, index i indicates the x-, y-, or z-component of the vector field, and u_i' and p' are the respective fluctuation components.

The Navier-Stokes equations are

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{4.23}$$

$$\rho\left(\frac{\partial U_i}{\partial t} + u_j\frac{\partial U_i}{\partial x_j}\right) = -\frac{\partial P}{\partial x_i}\delta_{ij} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \rho\overline{U_i'U_j'}\right] - \rho f_i \tag{4.24}$$

where $\rho\overline{U_i'U_j'}$ is called Reynolds stress tensor and can be further expressed as

$$-\rho\overline{U_i'U_j'} = \mu_T\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) \tag{4.25}$$

where $\mu_T = C_\mu \rho(k^2/\epsilon)$ is the eddy viscosity in the $k-\epsilon$ model. The turbulent kinetic energy k and turbulent dissipation rate ϵ are obtained by solving

$$\rho\left(\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_T}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\epsilon \tag{4.26}$$

$$\rho\left(\frac{\partial \epsilon}{\partial t} + U_j \frac{\partial \epsilon}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_T}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] + \frac{\rho\epsilon}{k}(C_1 P_k - C_2\epsilon) \tag{4.27}$$

where $P_k = \frac{\mu_T}{2}\left|\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right|^2$ is the production of turbulent kinetic energy. Eqs. (4.26) and (4.27) also contain empirical constants, such as $C_\mu = 0.09$, $C_1 = 1.44$, $C_2 = 1.92$, $\sigma_k = 1.0$ and $\sigma_\epsilon = 1.3$ [113]. To preclude the division by zero for k/ϵ in μ_T and ϵ/k in Eq. (4.27), an auxiliary parameter $\gamma = \epsilon/k$ is introduced [114]. μ_T is bounded by a lower limit $\mu_{\min}$ as a certain fraction of the dynamic viscosity $0 < \mu_{\min} \leq \mu$ and an upper limit $\mu_{\max} = \rho l_{\max}\sqrt{k}$, where $l_{\max}$ is the maximum admissible mixing length. The limited mixing length $l_\star$ is defined as

$$l_\star = \begin{cases} \rho\frac{C_\mu k^{\frac{3}{2}}}{\epsilon} & \text{if } \frac{C_\mu k^{\frac{3}{2}}}{\epsilon} < l_{\max} \\ \rho l_{\max} & \text{otherwise} \end{cases} \tag{4.28}$$

As a result, the modified μ_T and γ can be expressed as $\mu_T = \max\{\mu_{\min}, l_\star\sqrt{k}\}$ and $\gamma = (C_\mu k/\mu_T)\rho$.

4.3.2 Boundary conditions

The Dirichlet inflow boundary condition for u_i is

$$U_i = g_i^{(1)} \tag{4.29}$$

The Neumann inflow boundary condition for U_i is

$$-P\delta_{ij} + (\mu + \mu_T)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) = -g_i^{(2)} \tag{4.30}$$

The inflow boundary condition can be expressed in general as

$$\left[-P\delta_{ij} + (\mu + \mu_T)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] + hU_i = (h-1)g_i \tag{4.31}$$

The inflow boundary conditions for k and ϵ are

$$k = \frac{3}{2}(0.05|U_i|^2) \tag{4.32}$$

$$\epsilon = C_\mu^{\frac{3}{4}} \frac{k^{\frac{3}{2}}}{0.01} \tag{4.33}$$

The normal gradients of U_i, k, and ϵ at the outflow boundary are zeros. At other boundaries, wall functions are applied. The wall function for U_i is

$$(\mu + \mu_T)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) = \tau_s \tag{4.34}$$

where τ_s is the general shear stress applied on solid walls. The wall functions for k and ϵ are

$$\frac{\partial k}{\partial x_j} = 0 \tag{4.35}$$

$$\epsilon = \frac{\rho u_\tau^4}{\kappa_c y^+ \mu} \tag{4.36}$$

where the friction velocity $u_\tau = C_\mu^{\frac{1}{4}} k^{\frac{1}{2}}$, and the local Reynolds number $y^+ = \rho u_\tau y/\mu$. Here y denotes the width of the boundary layer, and κ_c =0.41 is the von Kármán constant. The value of y^+ needs to be within the range from 11.06 to 300.

4.3.3 Finite-element discretization of turbulent flow model

After introducing test functions Q and V_i, the weak formulation of Eqs. (4.23) and (4.24) over an element domain Ω^e can be expressed as

$$\int_{\Omega^e} Q \frac{\partial U_i}{\partial x_i}\, d\mathrm{x} = 0 \tag{4.37}$$

$$\int_{\Omega^e} \left\{ \rho\left(V_i \frac{\partial U_i}{\partial t} + V_i U_j \frac{\partial U_i}{\partial x_j}\right) + \frac{\partial V_i}{\partial x_j}\left[-P\delta_{ij} + (\mu + \mu_T)\left(\frac{\partial U}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] - \rho V f_i \right\} dx$$

$$-\int_{\partial\Omega_1} V_i(h-1)g_i ds + \int_{\partial\Omega_1} hV_iU_j ds - \int_{\partial\Omega_2} V_i\tau_s ds = 0 \tag{4.38}$$

respectively, where $\partial\Omega_1$ is the subdomain of the inflow boundary and $\partial\Omega_2$ is the subdomain of the solid walls. With shape functions ψ_u and ϕ_u, dependent variables U_i and P can be approximated as

$$U_i = {\psi_u}^T \boldsymbol{U}_i \tag{4.39}$$

$$P = \phi_u^T \boldsymbol{P} \tag{4.40}$$

By replacing V_i to ψ_u and Q to ϕ_p. Eqs. (4.37) and (4.38) can be reformulated as

$$-\mathbf{Q}^T\boldsymbol{U} = \mathbf{0} \tag{4.41}$$

$$\mathbf{M_u}\dot{\boldsymbol{U}} + \mathbf{C}(\boldsymbol{U})\boldsymbol{U} + \mathbf{K_u}\boldsymbol{U} - \mathbf{Q}\boldsymbol{P} = \boldsymbol{F}_\mathbf{u} \tag{4.42}$$

or in a matrix form as

$$\begin{bmatrix}\mathbf{M_u} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{U}}\\ \dot{\boldsymbol{P}}\end{bmatrix} + \begin{bmatrix}\mathbf{C}(\boldsymbol{U}) + \mathbf{K_u} & -\mathbf{Q}\\ -\mathbf{Q^T} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{P}\end{bmatrix} = \begin{bmatrix}\boldsymbol{F}_\mathbf{u}\\ \mathbf{0}\end{bmatrix} \tag{4.43}$$

where the coefficient matrices are defined as

$$\mathbf{M_u} = \int_{\Omega^e} \rho\psi_u\ {\psi_u}^T dx$$

$$\mathbf{C}(\boldsymbol{U}) = \int_{\Omega^e} \rho\boldsymbol{U}\psi_u\ \frac{\partial{\psi_u}^T}{\partial x_j} dx$$

$$\mathbf{K_u} = \int_{\Omega^e} (\mu + \mu_T)\frac{\partial\psi_u}{\partial x_i}\frac{\partial{\psi_u}^T}{\partial x_j} dx + \int_{\partial\Omega_1^e} h\psi_u\ {\psi_u}^{\mathbf{T}} ds$$

$$\mathbf{Q} = \int_{\Omega^e} \frac{\partial\psi_u}{\partial x_i}\phi_p^T dx$$

$$\boldsymbol{F}_\mathbf{u} = \int_{\Omega^e} \rho\psi_u\, f_i\, dx + \int_{\partial\Omega_1^e} \psi_u\,(h-1)g_i ds + \int_{\partial\Omega_2^e} \psi_u\,\tau ds \tag{4.44}$$

Eqs. (4.26) and (4.27) can be discretized in a similar way. With shape functions $\psi_\mathbf{k}$ and $\psi_\boldsymbol{\epsilon}$ defined, the matrix form of k equation is

$$\mathbf{M_k}\dot{\boldsymbol{k}} + \mathbf{K_k}\boldsymbol{k} = \boldsymbol{F}_\mathbf{k} \tag{4.45}$$

where

$$\mathbf{M_k} = \int_{\Omega^e} \rho \psi_\mathbf{k} \, \psi_\mathbf{k}^T dx,$$

$$\mathbf{K_k} = \int_{\Omega^e} \rho \boldsymbol{U} \psi_\mathbf{k} \frac{\partial \psi_\mathbf{k}^T}{\partial x_j} dx + \int_{\Omega^e} (\mu + \mu_T) \frac{\partial \psi_\mathbf{k}}{\partial x_i} \frac{\partial \psi_\mathbf{k}^T}{\partial x_j} dx + \int_{\Omega^e} \rho \gamma \psi_\mathbf{k} \psi_\mathbf{k}^\mathbf{T} \, dx$$

$$\boldsymbol{F_k} = -\int_{\Omega^e} \psi_\mathbf{k} P_k \, dx.$$

The matrix form of ϵ equation is

$$\mathbf{M_\epsilon} \dot{\boldsymbol{\epsilon}} + \mathbf{K_\epsilon} \boldsymbol{\epsilon} = \boldsymbol{F_\epsilon} \tag{4.46}$$

where

$$\mathbf{M_\epsilon} = \int_{\Omega^e} \rho \psi_\epsilon \, \psi_\epsilon^T dx,$$

$$\mathbf{K_\epsilon} = \int_{\Omega^e} \rho \boldsymbol{U} \psi_\epsilon \frac{\partial \psi_\epsilon^T}{\partial x_j} dx + \int_{\Omega^e} (\mu + \mu_T) \frac{\partial \psi_\epsilon}{\partial x_i} \frac{\partial \psi_\epsilon^T}{\partial x_j} dx + \int_{\Omega^e} \rho C_2 \gamma \psi_\epsilon \psi_\epsilon^\mathbf{T} \, dx$$

$$\boldsymbol{F_\epsilon} = -\int_{\Omega^e} \rho \gamma C_1 \psi_\epsilon P_k \, dx.$$

4.3.4 PBCS formulation for turbulent flow

The PBCS mechanism is established based on the finite-element discretization of the turbulent flow model. In the recovery step, the inverse problem is solved to find the load vector $\boldsymbol{F_u}$ in Eq. (4.43) based on some limited measurements of velocities $\boldsymbol{U}^*$. In the reconstruction step, the complete velocity field, turbulent kinetic energy and turbulent dissipation rate are reconstructed simultaneously by solving the forward problem based on the recovered load vector.

4.3.4.1 Steady state problem

For the steady state model, the time derivatives in the Navier-Stokes and $k-\epsilon$ equations are ignored. Therefore, Eq. (4.42) can be simplified as

$$\mathbf{C}(\boldsymbol{U})\boldsymbol{U} + \mathbf{K_u}\boldsymbol{U} - \mathbf{Q}\boldsymbol{P} = \boldsymbol{F_u} \tag{4.47}$$

Eq. (4.47) can be further linearized and rearranged to

$$\boldsymbol{U}_k - \boldsymbol{\alpha}\mathbf{Q}\boldsymbol{P}_{k-1} = \boldsymbol{\alpha}\boldsymbol{F_u} \tag{4.48}$$

where $\boldsymbol{\alpha} = (\mathbf{C}(\boldsymbol{U}_{k-1}) + \mathbf{K_u})^{-1}$. Velocity $\boldsymbol{U}_{k-1}$ and pressure $\boldsymbol{P}_{k-1}$ come from the previous iteration. The load vector $\boldsymbol{F}_\mathbf{u}$ is recovered from some measured velocities first. Multiplying $\boldsymbol{\Phi}$ to both sides Eq. (4.48) for the steady states, we have

$$\boldsymbol{\Phi U}_k - \boldsymbol{\Phi\alpha Q P}_{k-1} = \boldsymbol{\Phi\alpha F}_\mathbf{u} \tag{4.49}$$

The PBCS recovery is to find load vector $\boldsymbol{F}_\mathbf{u}$ such that the difference between the experimental measurements $\boldsymbol{U}^*$ and the selected measurements $\widetilde{\boldsymbol{U}} = \boldsymbol{\Phi U}_k$ from model is minimized as

$$\min\|\boldsymbol{U}^* - \boldsymbol{\Phi U}_k\|_2 \tag{4.50}$$

subject to the constraints in Eq. (4.48) and

$$\|\boldsymbol{F}_\mathbf{u}\|_0 < s_l \tag{4.51}$$

The algorithm to recover the load vector $\boldsymbol{F}_\mathbf{u}$ and reconstruct the steady state velocity field is shown in Table 3. Because Eq. (4.49) for the recovery and Eq. (4.43) for the reconstruction are tightly coupled with Eqs. (4.45) and (4.46), the *k*-ε equations also need to be solved simultaneously. The recovery and reconstruction steps become intertwined. The iterative process starts with the initial guess of the velocity field $\boldsymbol{U}_0$, the kinetic energy and the dispassion rate can be estimated by solving Eqs. (4.45) and (4.46). The load vector $\boldsymbol{F}_\mathbf{u}$ is then recovered with measurements based on Eqs. (4.48), (4.50) and (4.51). The recovery is based on the constrained OMP algorithm in Table 1. With the recovered load vector as well as the estimated kinetic energy and dispassion rate, the velocity and pressure fields are updated based on Eq. (4.43). A damping factor *c* between 0 and 1 is introduced to limit the change of each variable within a small range after each iteration so that the oscillation of solutions around the local optimum can be avoided. As a result, the convergence rate of solutions can be improved. For the steady-state problem, the velocity and pressure fields, the kinetic energy and the dissipation rate are obtained with the iterative process until the solution

converges.

Table 3. PBCS algorithm for steady-state turbulent flow

Input:	Measurements $\boldsymbol{U}^*$; measurement matrix $\boldsymbol{\Phi}$; model matrices $\boldsymbol{\alpha}$ and $\mathbf{Q}$. (or combined intrinsic matrix $\boldsymbol{\Theta}$); desired sparsity m.
Output:	Estimated load vector $\boldsymbol{F}_{\mathbf{u}}$. Complete velocity and pressure field, turbulent kinetic energy $\boldsymbol{k}$, and dissipation rate $\boldsymbol{\epsilon}$
Procedure:	1. Initialize the residual $\mathbf{r_0} = \boldsymbol{U}^*$; velocity $\boldsymbol{U}_0$, pressure $\boldsymbol{P}_0$, turbulent kinetic energy $\boldsymbol{k}_0$ and dissipation rate $\boldsymbol{\epsilon}_0$. 2. Estimate $\boldsymbol{k}_1$ and $\boldsymbol{\epsilon}_1$ by solving k and ϵ equations. 3. Recover the load vector $\boldsymbol{F}_{\mathbf{u}}$ with the constrained OMP algorithm 4. Estimate $\boldsymbol{U}_1$ and $\boldsymbol{P}_1$ by solving Navier-Stokes equations. 5. Update $\boldsymbol{U}_0 = c\boldsymbol{U}_0 + (1-c)\boldsymbol{U}_1$, $\boldsymbol{P}_0 = c\boldsymbol{P}_0 + (1-c)\boldsymbol{P}_1$, $\boldsymbol{k}_0 = c\boldsymbol{k}_0 + (1-c)\boldsymbol{k}_1$, $\boldsymbol{\epsilon}_0 = c\boldsymbol{\epsilon}_0 + (1-c)\boldsymbol{\epsilon}_1$, where c is the damping factor. 6. Return to step 2 until results converge.

4.3.4.2 Transient problem

Most turbulent flows are unsteady and time dependent. The PBCS mechanism to monitor the steady-state flow is just a special case for transient flow. For the transient mode, the general formulation in Eq. (4.42) can be rearranged as

$$\boldsymbol{U}_k - \boldsymbol{\alpha}(\mathbf{Q}\boldsymbol{P}_k + (\Delta t)^{-1}\mathbf{M}_{\mathbf{u}}\boldsymbol{U}_{k-1}) = \boldsymbol{\alpha}\boldsymbol{F}_{\mathbf{u}} \tag{4.52}$$

where $\boldsymbol{\alpha} = \left((\Delta t)^{-1}\mathbf{M}_{\mathbf{u}} + \mathbf{K}_{\mathbf{u}} + \mathbf{C}(\boldsymbol{U}_{k-1})\right)^{-\mathbf{1}}$ and Δt is the time step. The pressure $\boldsymbol{P}_k$ can be approximated as $\boldsymbol{P}_k = \boldsymbol{P}_{k-1} + (\boldsymbol{P}_{k-1} - \boldsymbol{P}_{k-2})$. That is, $\boldsymbol{P}_k$ is approximated by extrapolating from

the previous two time steps. We have constraint in Eq. (4.52) for the recovery process. The velocity and pressure fields, the kinetic energy and the dissipation rate are updated based on time step Δt in Table 4.

As time step Δt is reduced to improve the reconstruction accuracy, the computation becomes more expensive. Because the accuracy of reconstructed results at the current time step also depends on reconstructed results at the previous time step. More measurements can improve the accuracy but also increase the computational cost. The location of sensors can affect reconstruction results for both steady-state and transient cases. Therefore, the locations of sensors need to be optimized. Given the availability of experimental data to test the PBCS mechanisms, only the special case of steady-state turbulent flow is demonstrated and evaluated. The general transient case will be tested in future work.

Table 4. PBCS algorithm for transient turbulent flow

Input:	Measurements $\boldsymbol{U}^*$; measurement matrix $\boldsymbol{\Phi}$; model matrices $\boldsymbol{\alpha}$ and $\mathbf{Q}$. (or combined intrinsic matrix $\boldsymbol{\Theta}$); desired sparsity m. Time step Δt.
Output:	Estimated load vector $\boldsymbol{F}_{\mathbf{u}}$. Complete velocity and pressure field, turbulent kinetic energy $\boldsymbol{k}$, and dissipation rate $\boldsymbol{\epsilon}$
Procedure:	1. Initialize the residual $\mathbf{r_0} = \boldsymbol{U}^*$, the time $t = 0$, velocity $\boldsymbol{U}_{\mathrm{t=0}}$, pressure $\boldsymbol{P}_{\mathrm{t=0}}$, turbulent kinetic energy $\boldsymbol{k}_{\mathrm{t=0}}$ and dissipation rate $\boldsymbol{\epsilon}_{\mathrm{t=0}}$. 2. Estimate $\boldsymbol{k}_{t+\Delta t}$ and $\boldsymbol{\epsilon}_{t+\Delta t}$ by solving k and ϵ equations. 3. Recover the load vector $\boldsymbol{F}_{\mathbf{u}}$ with the constrained OMP algorithm 4. Estimate $\boldsymbol{U}_{t+\Delta t}$ and $\boldsymbol{P}_{t+\Delta t}$ by solving Navier-Stokes equations. 5. Return to step 2 and $= t + \Delta t$.

4.4 Demonstration

The backward-facing step flow shown in Figure 30 is used to demonstrate and evaluate the performance of the PBCS mechanism for steady-state flows. The experimental data are obtained from the work of Driver and Seegmiller [115]. The height of the backward-facing step, H, is 1.27 cm. Other relative dimensions are also shown in Figure 30. A streamline velocity of 44.2 m/s is imposed at the inlet and used as the reference velocity u_{ref}. The wall boundary-layer thickness is 1.9 cm, and the momentum thickness Reynolds number is 5000 at a location of $4H$ upstream of the step. The division streamline, where the streamwise velocity is zero, is indicated as the dash line in Figure 30. The recirculating flow occurs under the division streamline. The complete experimental measurements were taken at the locations indicated as dots in Figure 31. Velocities were measured with a two-color LDA. Missed measurements on boundaries, such as the ones at

$y = H$, $y = 9H$ and $x = 0$, are predicted by linearly extrapolating two adjacent measurements.

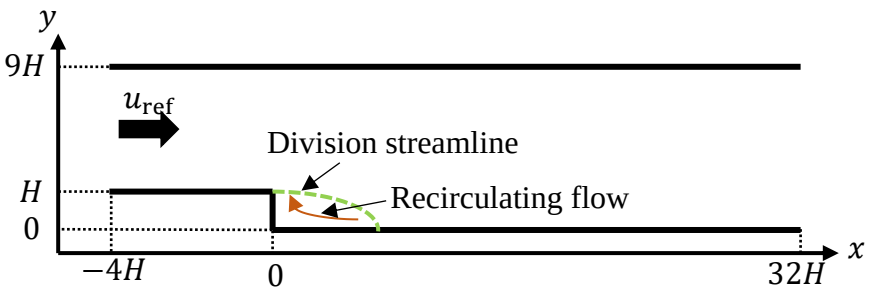


Figure 30. The setup of the backward-facing step flow

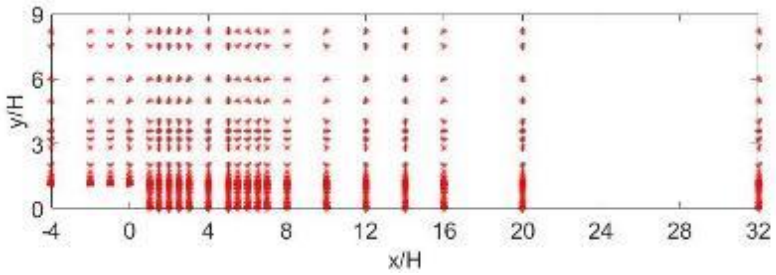


Figure 31. Complete experimental measurement points

The selected measurement locations for PBCS are indicated by dash lines in Figure 32. Velocities are measured near the top boundary and the boundaries near inlet. Near the bottom boundary, the gradient of velocity is high. Therefore, velocities in the shaded region with a height of L are also measured. Measurements along the dash lines and in the shaded region are used to recover the load vector $\boldsymbol{F}_{\mathbf{u}}$ in Eq.(4.52). Different heights of the shaded region were chosen to analyse the performance of the PBCS algorithm, which is sensitive to the number of measurements.

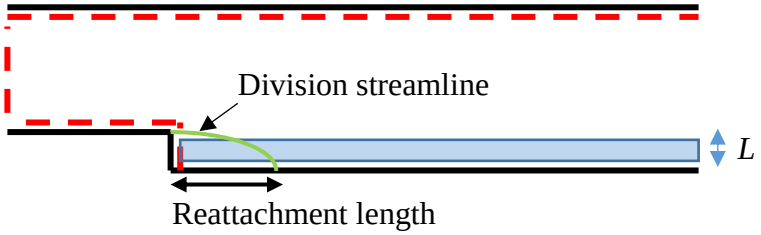


Figure 32. Measurement locations for PBCS

The mesh model of the domain is shown in Figure 33. The meshes at the top boundary, bottom boundary, and the region of recirculating flow are locally refined. The mesh directly above and adjacent to the step is not refined because the mesh is fine enough in this region and further refinement does not improve the accuracy of the result while increasing the computational cost. The maximum mesh size is 4.21 mm and the total number of nodes is 7889.

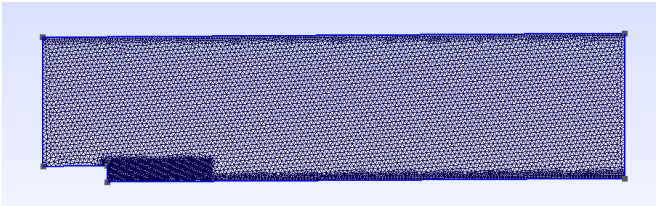


Figure 33. Mesh of the domain

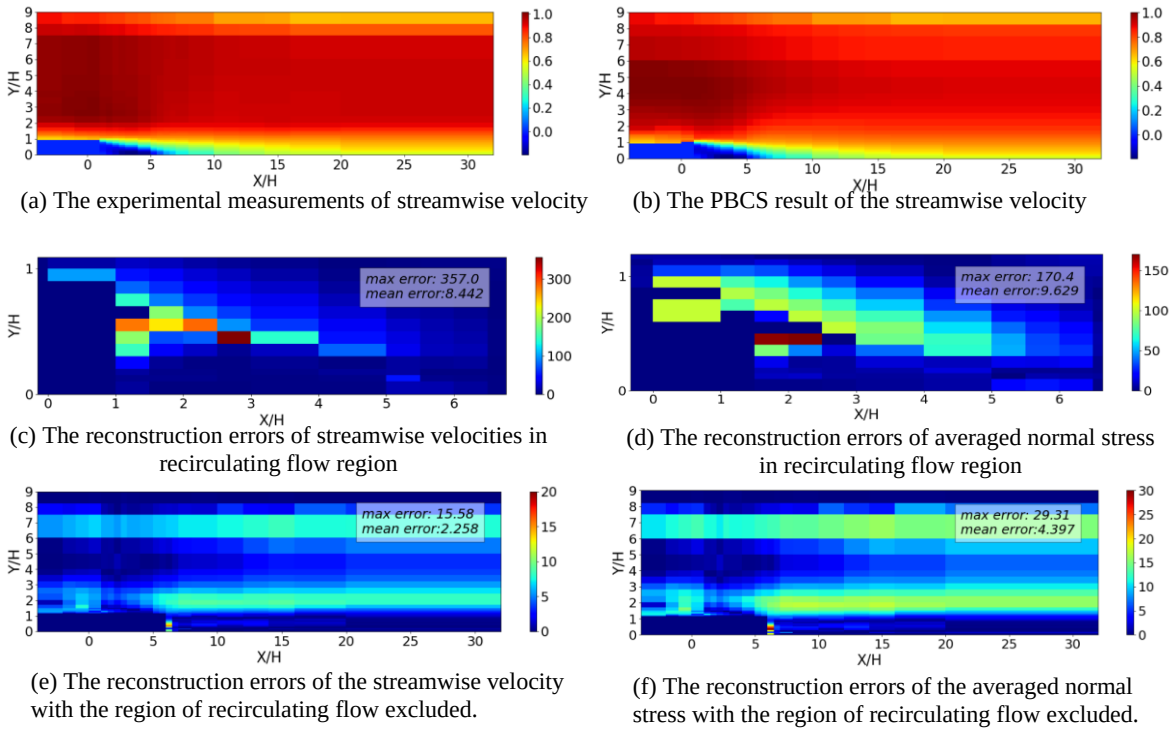
Not all nodal velocities near the boundaries denoted by dash lines and in the shaded region in Figure 32 can be directly measured with the limitations of sensor accessibility or sampling resolution. From two adjacent measurements, the nodal velocities near the top and left boundaries can be estimated by linear interpolation. The ones in the shaded region are also estimated with the 2D interpolation of four nearby measurements in the horizontal and vertical directions. Those measured and estimated nodal velocities are used to recover the load vector in Eq. (4.52).

The velocities from the PBCS reconstruction and experimental results are compared. Three

cases with different sizes of the shaded region are tested. The height L in Figure 32 is chosen to be $0.1H$, $0.3H$, and $0.5H$ respectively. In these three cases, there are 89, 140 and 174 actual measurements respectively used for recovery. Thus the corresponding compression ratios are 7889/89=88.6, 7889/140=56.4, and 7889/174=45.3.

When L=$0.3H$, the experimental and PBCS results after mapping to the coordinates of measured locations in Figure 31 are shown in Figure 34 (a) and (b) respectively, where the streamwise velocities u_x/u_{ref} are shown. The relative error of streamwise velocities in the recirculating flow region are shown in Figure 34 (c). The relative error of streamwise velocity is computed as $error_{u_x} = |u_x^{recons} - u_x^{exp}\ |/u_x^{exp} \times 100\%$, where u_x^{recons} is the streamwise velocity after reconstruction and u_x^{exp} is the streamwise velocity from experimental measurements. Some interpolation is necessary when the locations of the nodes and experimental measurements do not exactly match. The relative errors of the averaged normal stresses $0.5\left(u_x^2 + u_y^2\right)/u_{ref}^2$ [115,116,117] in the recirculating flow region are also shown in Figure 34 (d). It is found that most errors come from the region of recirculating flow. This is because the standard $k-\epsilon$ model may not be able to predict the accurate velocity field in this region [117]. The interpolation of nodal values in this region after reconstruction can also introduce additional errors because of the large gradients. The velocities near the division streamline have small absolute values, which can also result in large relative errors. If the region of recirculating flow is excluded, the relative errors for the rest of the domain are much smaller. The relative errors of streamwise velocities and averaged normal stresses with the region of the recirculating flow excluded are shown in Figure 34 (e) and (f) respectively. The reconstructed averaged normal stresses show larger errors than the streamwise velocities because the vertical velocities have small absolute values but high gradients, which also results in larger interpolation errors. The computational time for one iteration of

recovery and reconstruction is 41.3s on a single-CPU desktop computer (2.0 GHz), and it takes 140 iterations to converge.



(a) The experimental measurements of streamwise velocity

(b) The PBCS result of the streamwise velocity

(c) The reconstruction errors of streamwise velocities in recirculating flow region

(d) The reconstruction errors of averaged normal stress in recirculating flow region

(e) The reconstruction errors of the streamwise velocity with the region of recirculating flow excluded.

(f) The reconstruction errors of the averaged normal stress with the region of recirculating flow excluded.

Figure 34. Comparison of measured and predicted results when L =0.3H.

In the PBCS algorithm, computing $\boldsymbol{\alpha}$ in Eq. (4.52) directly by inverting the square matrix can be computationally expensive for high-fidelity models with a large number of nodes. Instead, $\boldsymbol{\Phi\alpha}$ is computed together. Since $\boldsymbol{\Phi}$ has the size of $M \times N$ and $M \ll N$ if the compression ratio is high, computing $\boldsymbol{\Phi\alpha}$ only needs M rows of $\boldsymbol{\alpha}$ since there are only a limited number of measurements. Denote $\boldsymbol{\beta} = \boldsymbol{\alpha}^{-1}$ and $\boldsymbol{\alpha\beta} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Calculating $\boldsymbol{\beta}$ is equivalent to solve $\boldsymbol{\alpha\beta}_i = \mathbf{e}_i, i = 1, \dots, N$, where $\boldsymbol{\beta}_i$ is the i^{th} column of $\boldsymbol{\beta}$ and $\mathbf{e}_i$ is the i^{th} column of $\mathbf{I}$. With the notation $\boldsymbol{\omega} = \boldsymbol{\Phi\alpha}$, $\boldsymbol{\omega}$ can be found by solving $[\mathbf{C}(\mathbf{u}_{k-1}) + \mathbf{K}_\mathbf{u}]^\mathbf{T}\boldsymbol{\omega}^\mathbf{T} = \boldsymbol{\Phi}^\mathbf{T}$ in the steady state case. Since matrices $[\mathbf{C}(\mathbf{u}_{k-1}) + \mathbf{K}_\mathbf{u}]^\mathbf{T}$ and $\boldsymbol{\Phi}^\mathbf{T}$ are sparse, the computation of $\boldsymbol{\omega}$ can be much more efficient when $M \ll N$. For this 2D turbulent flow problem with a total of 7889 mesh nodes, the

computation of $\boldsymbol{\alpha}$ requires the inverse of a matrix with the size of 15778×15778. The computational time for $\boldsymbol{\alpha}$ along was 214.56s in our test. In our new method, $\boldsymbol{\omega}$ is computed with $M = 1352$ and $N = 15778$ if 1352 nodal values are from measurements. The computational time for $\boldsymbol{\omega}$ is only 7.81s. Therefore, the computational cost has been significantly reduced in our method.

The reattachment length is an important factor to consider when comparing the PBCS reconstruction and actual experimental results for the backward-facing step flow. The reattachment length is defined as the distance from the step to where the flow resumes in the positive flow direction, as shown in Figure 32. The reattachment length from the PBCS reconstruction has a relative error of 3.5% in comparison with the actual experimental measurement. The error of the reattachment length is small, because the measurements near the boundary are used in the recovery.

The reconstruction results and errors for L=0.1H and L=0.5H are similar to those in the case of L=0.3H. They are summarized as follows. The averaged errors for the streamwise velocity after excluding the region of recirculating flow are 3.41%, 2.26%, and 2.25% for the cases of L=0.1H, L=0.3H, and L=0.5H respectively. The averaged errors for the averaged normal stresses after excluding the region of recirculating flow are 6.6%, 4.4%, and 4.36% respectively. It is seen that more measurements in the high gradient region help reduce the PBCS error. For all cases, the large errors of the streamwise velocities appear in the region where y=7H. This is mainly due to extrapolation error. The nodal velocities at the top boundary $y = 9H$ are used for the recovery and reconstruction. However, they were not directly measured. They were obtained by linear extrapolations based on the two immediate rows of sparse measurements below $y = 9H$, as shown in Figure 31. The extrapolation can introduce large errors. If measurements on boundaries at $y = 9H$ are directly available, the error can be further reduced. The errors of the streamwise velocities

in the region of $y = 2H$ are also high. This is because the gradients of the velocities are high in this region, which causes larger interpolation errors. The errors near the walls and directly above the step are much smaller, because more measurements in these regions were used for recovery and reconstruction. When more direct measurements are used, the complete velocity field can be reconstructed more accurately but with a higher computational cost.

In a further test, more measurements in the region of recirculating flow are taken to improve the sensing accuracy, since this region is usually of interest. As shown in Figure 35, the two shaded regions require enough measurement samples for accurate PBCS reconstruction. The left one is the region of recirculating flow, where more samples are taken in comparison with Figure 32. In this test, L=0.2H. A total of 179 measurements of velocities are taken, including the boundaries and two shaded regions. The compression ratio is 7889/179=44.1, which is close to the previous case of L =0.5H. The comparisons between experimental and reconstructed results are shown in Figure 36. In Figure 34 (b) and Figure 36 (b), there is a jump of the streamwise velocity at the end of the step because the streamwise velocity is set to be 0 at the location of X=0 and Y=1H. It was found that the turbulent kinetic energy solved by the PBCS algorithm does not converge near the recirculating flow region if the streamwise velocity at X=0 and Y=1H is non-zero. This numerical stability issue mostly comes from the standard k-ε model that we use. The mesh near this region of sharp corner is also not dense enough. The average and maximum relative errors of streamwise velocity are reduced to 4.01% and 66.4% respectively, from previously 4.09% and 139.9 % when L =0.5H. In the new test, the largest error occurs in the narrow region near the division streamline, because of the small absolute values of velocities. The mesh size of this narrow region is also not fine enough to allow the locations between nodes and measurements to match exactly. Nodal velocity values near the division streamline are approximated by the linear interpolation from the

measurements. Very fine mesh will be needed in this narrow region, which however can significantly increase computational expenses. When the velocities in the narrow region near the division streamline are excluded, the errors are shown in Figure 36 (e) and (f). The average error is 3.16% for the streamwise velocity and 6.17% for the averaged normal stresses.

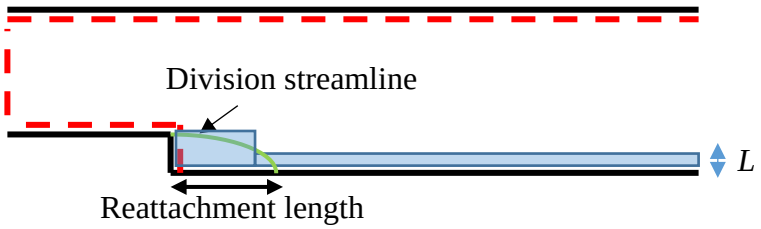
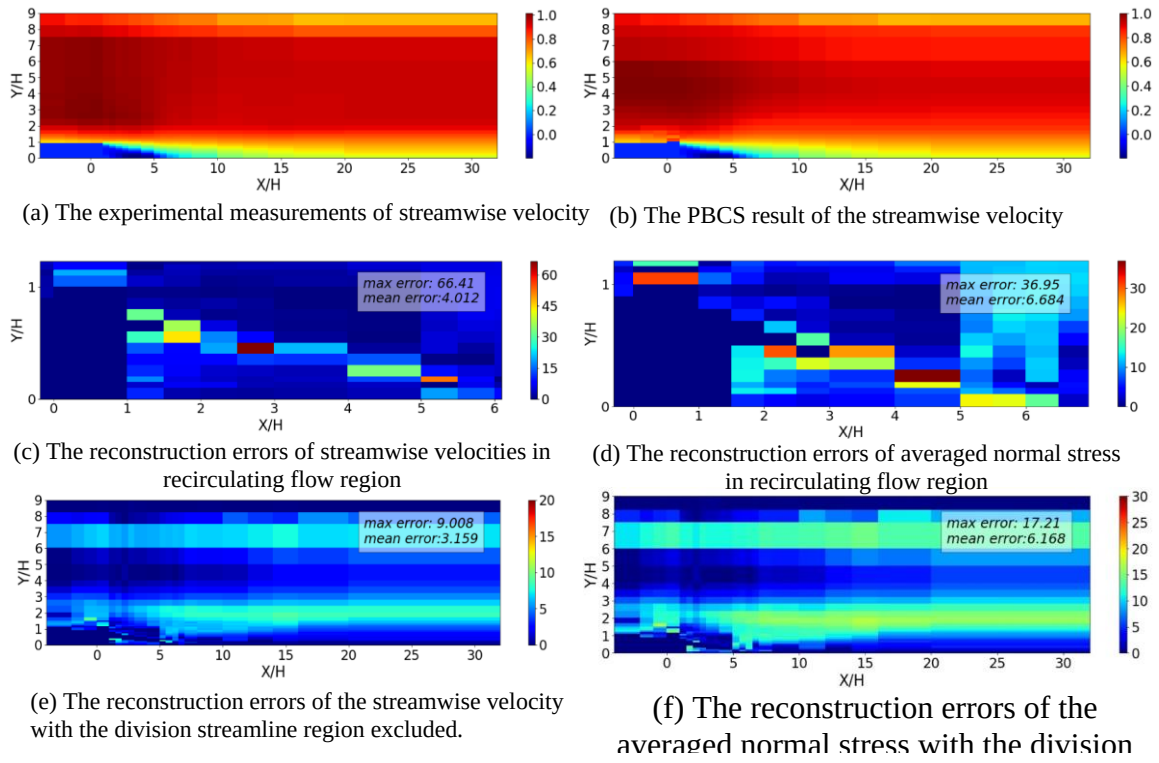


Figure 35. Measurement locations if the region of the recirculating flow is of interest



(a) The experimental measurements of streamwise velocity

(b) The PBCS result of the streamwise velocity

(c) The reconstruction errors of streamwise velocities in recirculating flow region

(d) The reconstruction errors of averaged normal stress in recirculating flow region

(e) The reconstruction errors of the streamwise velocity with the division streamline region excluded.

(f) The reconstruction errors of the averaged normal stress with the division

Figure 36. Comparison of measured and predicted results when L =0.2H and velocities in the region of recirculating flow are also measured.

4.5 Discussions

The major benefit of PBCS is that it provides the flexibility for measurement locations. When some regions are not directly accessible by sensors, the measurements at other locations can compensate. When some regions are more important than others, sensors can be relocated to increase the accuracy while high compression ratio and sampling efficiency can be maintained. The experimental tests in this work show that decisions of sample locations can be made to improve both accuracy and efficiency. When more measurements are taken in the regions of interest while fewer are taken in other regions, the overall reconstruction accuracy can be improved while maintaining a similar compression ratio.

The major source of PBCS errors is the physical model of the system to be monitored, as a result of approximation, simplification, and numerical treatment. Some additional procedures for PBCS error estimation and compensation can be helpful to reduce the sensing error. For instance, in CHAPTER 3, a Gaussian process regression model was built to predict the errors in both spatial and temporal domains. After training, the model can be applied to predict errors, and the error compensation helped reduce the PBCS reconstruction error significantly.

The major technical challenge for the application of PBCS to monitor the turbulent flow, especially real-time experimental measurements, is the computational efficiency. Particularly calculating the inverse of matrices is needed in the PBCS algorithm. Although the method shown in Section 4 has significantly improved the efficiency of inverse matrix calculation, the cost is still high for the real-time monitoring because of the repeated calculations. For transient flows, the choice of time step also affects the efficiency and accuracy of reconstruction. In CHAPTER 3, we have developed several methods to improve the computational efficiency, including the domain decomposition method and the constrained OMP algorithm. In future work, we will further evaluate the PBCS mechanism for transient turbulent flows with real-time experimental

measurements.

4.6 Conclusion

PBCS approaches are proposed to monitor laminar and transient velocity fields based on the Navier-Stokes equations and the RANS equations respectively. With a limited number of measurements, PBCS can be used to reconstruct the complete velocity fields more efficiently than traditional CS. The demonstrated compression ratio for the 2D turbulent flow can be as high as 40 to 80. In contrast, traditional CS only has a compression ratio of 3 to 5.

The PBCS mechanism was demonstrated with the steady-state backward-facing step flow, and its performance was compared with the experimental data from LDA measurement. Sensitivity studies show that the accuracy of PBCS reconstruction relies on the number of measurements used for recovery. If more measurements are used, the reconstruction accuracy is higher, yet also with higher computational cost. Therefore, the regions of interest, especially with high gradients, require more measurements if possible to maintain a reasonable accuracy level. The trade-off between the reconstruction accuracy and the computational cost is needed.

Future studies will include PBCS formulations with other turbulent fluid models in two or three dimensions. The PBCS framework for the laminar velocity field monitoring will also be demonstrated with experimental results. More extensive comparisons between the PBCS mechanisms and traditional measurement methods are needed for a more comprehensive assessment of accuracy and efficiency. The locations of sensing or sampling can affect the reconstruction accuracy for both steady-state and transient cases. Therefore, the locations of sensors need to be optimized to improve the reconstruction performance in future work.

CHAPTER 5. Metal Additive Manufacturing Process Monitoring

5.1 Introduction

In this Chapter, the temperature and velocity fields in the SLM process are monitored with a proposed PBCS approach based on a multi-physics model. In this model, heat transfer and fluid flow in the melt pool are tightly coupled. The multi-physics model is discretized with the finite element method. Based on some temperature measurements on the top surface of the build, the sparse load vector of the heat transfer model is recovered by solving an inverse problem. The complete temperature distribution is then reconstructed based on the prediction of the heat transfer model. With the complete temperature distribution, the boundary conditions for the fluid flow model and the complete velocity field are reconstructed based on the tightly coupled thermofluid model. The boundary conditions for the fluid flow model can also be recovered with velocity measurements in experiments. With the proposed PBCS approach, the high-resolution internal temperature and velocity fields in the melt pool can be obtained simultaneously. When there are limitations in sensor placement or accessibility, or when the resolution of sensors used for data collection is low, PBCS can help obtain high-fidelity information and improve the efficiency of process monitoring.

In Section 5.2, the thermofluid model of the melt pool and its finite-element discretization are described. Different factors of the thermofluid model such as the dimension of the model, the mesh size and solver techniques are calibrated with experimental results. In Section 5.3, the PBCS mechanism is established based on the finite-element discretization of the multi-physics model. The two PBCS formulations of reconstruction based on temperature measurements and velocity measurements are implemented. The proposed PBCS framework is then demonstrated and validated with experimental measurements in Section 5.4. The load vector in the heat transfer

model is first recovered and the complete temperature and velocity fields can be reconstructed based on the forward modelling.

5.2 Thermofluid model of melt pool

5.2.1 Formulation of the thermofluid model

The multi-physics model of the melt pool is developed by coupling heat transfer and fluid flow models. The differential equations for the conservation of mass, momentum, and energy are given as

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{5.1}$$

$$\rho\left(\frac{\partial u_i}{\partial t} + v_j\frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i}\delta_{ij} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + f_i \tag{5.2}$$

$$\rho\frac{C_p dT}{dt} + \rho C_p u_j \cdot \frac{\partial T}{\partial x_j} - \kappa\frac{\partial}{\partial x_i}\cdot\left(\frac{\partial T}{\partial x_j}\right) = 0 \tag{5.3}$$

where u_i is the velocity component in the i-th direction, ρ is the density, f_i is the volumetric force, p is the pressure, μ is dynamic viscosity, C_p is the specific heat at constant pressure, κ is the thermal conductivity, T is the temperature, and δ_{ij} is given as

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{5.4}$$

The evolution of the melt pool involves phase change. The liquid-solid domain is characterized as pure solid, pure liquid, and a mixture of solid and liquid (mushy zone). The material properties of the mushy zone can be estimated as

$$\rho_m = f_s\rho_s + f_l\rho_l \tag{5.5}$$

$$k_m = f_s k_s + f_l k_l \tag{5.6}$$

$$\mu_m = \mu_l\frac{\rho_m}{\rho_l} \tag{5.7}$$

where ρ_s and k_s are density and thermal conductivity in the solid phase, and $\rho_l$and k_l are those in the liquid phase, respectively. μ_l is the dynamic viscosity in the liquid phase. f_s and f_l are the solid and liquid mass fractions, which can be expressed as

$$f_l = \begin{cases} 1 & T > T_l \\ \frac{T-T_s}{T_l-T_s} & T_s \leq T \leq T_l \\ 0 & T < T_s \end{cases} \tag{5.8}$$

$$f_s = 1 - f_l \tag{5.9}$$

where T_s and T_l are solidus and liquidus temperatures, respectively. The apparent heat capacity method is used by including the latent heat as an additional term in the heat capacity. The effective heat capacity can be expressed as

$$C_{pm} = \frac{1}{\rho_m}\left(f_s\rho_s C_{ps} + f_l\rho_l C_{pl}\right) + L\frac{\partial \alpha_m}{\partial T} \tag{5.10}$$

where L is the latent heat and

$$\alpha_m = \frac{1}{2}\frac{f_s\rho_s - f_l\rho_l}{f_l\rho_l + f_s\rho_s}. \tag{5.11}$$

Boundary conditions are required for the forward modelling. For the heat transfer model, heat flux from the laser source, convection, and radiation is applied on the top surface, which can be represented as

$$q''_{in\_top} = q''_{LS} - h_c(T - T_\infty) - \sigma\varepsilon_s(T^4 - T_\infty^4) \tag{5.12}$$

where h_c is the heat transfer coefficient, T_∞ is the ambient temperature, and σ and ε_s represent the Stefan-Boltzmann constant and the emissivity respectively. The laser source is assumed to follow a Gaussian distribution as

$$q''_{LS} = \frac{2P_w\eta}{\pi R_L}\exp\left(-\frac{2(x^2+y^2)}{R_L^2}\right) \tag{5.13}$$

where P_w is the laser power, η is the laser absorbability, and R_L is the effective laser beam radius.

Additional volumetric buoyancy force f_b and damping force f_D are added to the conservation of momentum. f_b is the buoyancy force due to the difference of the densities in the melt pool, and expressed as

$$f_b = \rho_r g[1 - \beta(T - T_r)] \tag{5.14}$$

where ρ_r and T_r are reference density and temperature, respectively, g is the gravity field, and β is the thermal expansion coefficient. f_D is a Darcy term representing the damping force when fluid passes through a porous medium and is formulated as

$$f_D = -\frac{\mu_m}{A} u_i \tag{5.15}$$

where A is the isotropic permeability and can be expressed as

$$A^{-1} = \frac{A_0^{-1}(1-f_l)^2}{f_l^3+\tau} \tag{5.16}$$

where A_0 is a constant determined by the morphology of the mushy zone, and τ is a small number to avoid the singularity. For the pure solid, A approaches zero. For the pure liquid, A approaches to infinity.

For the fluid flow model, capillary force

$$\sigma_n = -\gamma \hat{n} \kappa_c \tag{5.17}$$

and Marangoni force

$$\sigma_t = \frac{d\gamma}{dT} \nabla_s T \tag{5.18}$$

are applied on the top surface, where γ is the surface tension, $\hat{n}$ is the normal direction, κ_c is the face curvature, and ∇_s is the gradient in the tangent plane. The capillary force acts in the normal direction, whereas the Marangoni force acts in the direction tangent to the surface. When the laser beam has low power density, depression is not formed in the melt pool. The dominating driving force is Marangoni force, which drives the melt flow from hotter region to cooler region [58].

5.2.2 Finite-element discretization

By introducing shape functions θ, ψ and ϕ for the vector of nodal temperatures $\boldsymbol{T}$, velocity components $\boldsymbol{u_i}$ and pressures $\boldsymbol{p}$ respectively, dependent variables T, u_i and p can be approximated as

$$T = \theta^T \boldsymbol{T},\ u_i = \psi^T \boldsymbol{u_i},\ p = \phi^T \boldsymbol{p} \tag{5.19}$$

With test functions w_1, w_2 and w_3, the weak formulation of Eqs. (5.1-5.3) over an element domain Ω^e is

$$\int_{\Omega^e} w_1 \frac{\partial u_i}{\partial x_i} d\Omega^e = 0 \tag{5.20}$$

$$\int_{\Omega^e} w_2 \left\{ \rho \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) + \frac{\partial p}{\partial x_i} \delta_{ij} - \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] - f_i \right\} d\Omega^e = 0 \tag{5.21}$$

$$\int_{\Omega^e} w_3 \left[\rho \frac{C_p dT}{dt} + \rho C_p u_j \cdot \frac{\partial T}{\partial x_j} - \kappa \frac{\partial}{\partial x_i} \cdot \left(\frac{\partial T}{\partial x_j} \right) \right] d\Omega^e = 0 \tag{5.22}$$

Test functions w_1, w_2 and w_3 can be replaced with ϕ, ψ and θ respectively. By substituting Eq. (5.19) into Eqs. (5.20-5.22), finite element equations including boundary conditions over an element domain Ω^e can be represented as

$$\left[\int_{\Omega^e} \phi \frac{\partial \psi^T}{\partial x_i} d\Omega^e \right] \boldsymbol{u_i} = 0 \tag{5.23}$$

$$\left[\int_{\Omega^e} \rho \psi \psi^T d\Omega^e \right] \dot{\boldsymbol{u}}_{\boldsymbol{i}} + \left[\int_{\Omega^e} \rho \psi (\psi^T v_j) \frac{\partial \psi^T}{\partial x_j} d\Omega^e \right] \boldsymbol{u_i} + \left[\int_{\Omega^e} \mu \frac{\partial \psi}{\partial x_j} \frac{\partial \psi^T}{\partial x_j} d\Omega^e \right] \boldsymbol{u_i} +$$

$$\left[\int_{\Omega^e} \mu \frac{\partial \psi}{\partial x_j} \frac{\partial \psi^T}{\partial x_i} d\Omega^e \right] \boldsymbol{u_j} - \left[\int_{\Omega^e} \frac{\partial \psi}{\partial x_i} \phi^T d\Omega^e \right] \boldsymbol{p} = \int_{\Omega^e} f_b \psi d\Omega^e + \int_{\Omega^e} f_D \psi d\Omega^e + \int_{\partial\Omega^e} \sigma_n \psi d(\partial\Omega^e) +$$

$$\int_{\partial\Omega^e} \sigma_t \psi d(\partial\Omega^e) \tag{5.24}$$

$$\left[\int_{\Omega^e} \rho C_p \theta \theta^T d\Omega^e \right] \dot{\boldsymbol{T}} + \left[\int_{\Omega^e} \rho C_p \theta (\psi^T u_j) \frac{\partial \theta^T}{\partial x_j} d\Omega^e \right] \boldsymbol{T} + \left[\int_{\Omega^e} \kappa \frac{\partial \theta}{\partial x_i} \frac{\partial \theta^T}{\partial x_j} d\Omega^e \right] \boldsymbol{T} =$$

$$\int_{\partial\Omega^e} q''_{in_{top}} \theta d(\partial\Omega^e) \tag{5.25}$$

where $\partial\Omega$ is the subdomain of the top surface.

Eqs. (5.23-5.25) can be reformulated as

$$\mathbf{Q}^T\boldsymbol{u} = \mathbf{0} \tag{5.26}$$

$$\mathbf{M_u}\dot{\boldsymbol{u}} + \mathbf{C}(\boldsymbol{u})\boldsymbol{u} + \mathbf{K_u}\boldsymbol{u} - \mathbf{Q}\boldsymbol{p} = \boldsymbol{F_u} \tag{5.27}$$

$$\mathbf{M}\dot{\boldsymbol{T}} + \mathbf{D}(\boldsymbol{u})\boldsymbol{T} + \mathbf{K}\boldsymbol{T} = \boldsymbol{L} \tag{5.28}$$

where the coefficient matrices are defined as

$$\mathbf{M_u} = \int_{\Omega^e} \rho\psi \ \psi^T d\Omega^e$$

$$\mathbf{C}(\boldsymbol{u}) = \int_{\Omega^e} \rho\psi(\psi^T u_j)\frac{\partial\psi^T}{\partial x_j} d\Omega^e$$

$$\mathbf{K_u} = \int_{\Omega^e} \mu\frac{\partial\psi}{\partial x_j}\frac{\partial\psi^T}{\partial x_j} d\Omega^e$$

$$\mathbf{Q} = \int_{\Omega^e} \frac{\partial\psi}{\partial x_i}\phi^T d\Omega^e$$

$$\boldsymbol{F_u} = \int_{\Omega^e} f_d\psi d\Omega^e + \int_{\Omega^e} f_D\psi d\Omega^e + \int_{\partial\Omega^e} \sigma_n\psi d(\partial\Omega^e) + \int_{\partial\Omega^e} \sigma_t\psi d(\partial\Omega^e)$$

$$\mathbf{M} = \int_{\Omega^e} \rho C_p\theta\theta^T d\Omega^e$$

$$\mathbf{D}(\boldsymbol{u}) = \int_{\Omega^e} \rho C_p\theta(\psi^T u_j)\frac{\partial\theta^T}{\partial x_j} d\Omega^e$$

$$\mathbf{K} = \int_{\Omega^e} \kappa\frac{\partial\theta}{\partial x_i}\frac{\partial\theta^T}{\partial x_j} d\Omega^e$$

$$\boldsymbol{L} = \int_{\partial\Omega^e} q''_{in_{top}}\theta d(\partial\Omega^e) \tag{5.29}$$

5.2.3 Calibration and validation of the physical model

The accuracy of the physical model is sensitive to different factors such as the dimension of the model, the mesh size and solver techniques. Therefore, before applying the developed multi-physics model to the proposed PBCS framework, the physical model is calibrated and validated

by experimental results from traditional measurement methods. The FEniCS is used to implement the finite-element formulation. A 2D model of the cross-section of melt pool is used instead of 3D model in order to reduce the computational cost. The size of the model is 1.5mm×0.3mm. The uniform mesh with the size of 0.0053mm is applied. Boundary conditions such as capillary and Marangoni forces, and the heat flux from the laser source, convection and radiation are applied on the top edge of the 2D domain. The segregated approach is used to solve the physical model, which can save the memory usage. The Krylov subspace solver [118] with block diagonal preconditioner [119] is applied to solve the fluid flow model in Eqs. (5.26) and (5.27). The MUMPS solver is used to solve the heat transfer model in Eq. (5.28). The selected solver techniques can be used for parallel computing.

The experimental results of temperature distribution obtained from Hooper's work [47] are used for calibration. The temperature distribution on the top surface of melt pool with a range between 1000 K and 4000 K for a single track was recorded for every 10 μs. One of the frames is shown in Figure 37. The temperature readings along the centreline are extracted to compare with the forward modelling prediction of the developed thermofluid model.

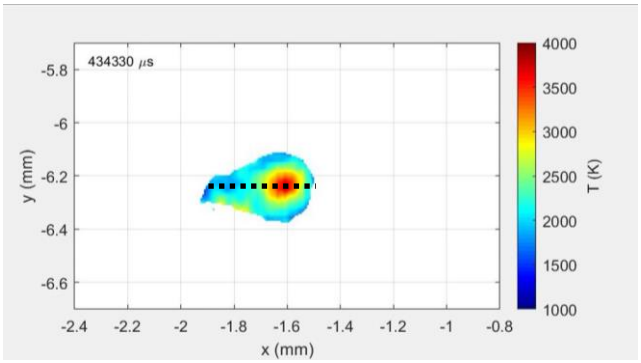


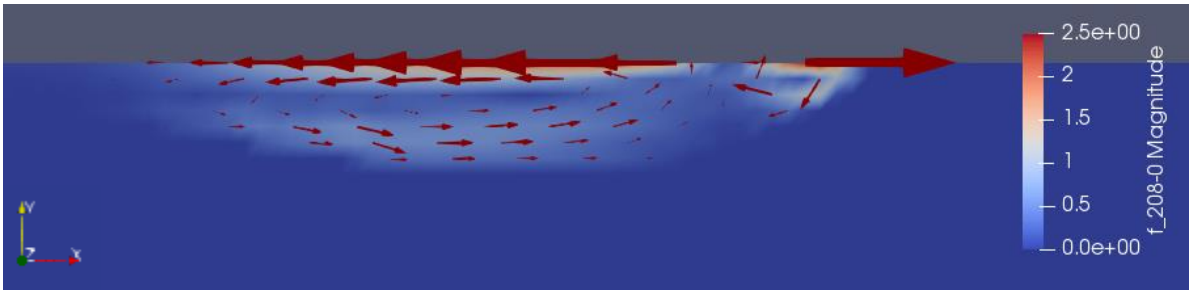
Figure 37. 2D temperature field of SLM melt pool top surface measured with high-speed imaging thermography [47]

The material used in the experiment is Ti6Al4V. The laser power and effective beam radius are 85 W and 32.5 μm respectively. The scanning speed of laser source is 0.87 m/s. The generated

temperature and velocity fields from the forward modelling are shown in Figure 38. The laser source moves along the positive x-direction. The maximum temperature is close to 3800 K, and the initial temperature of the model is 300 K. The temperature range of the color bar is set to be above 1500 K so that it is easier to visualize the temperature gradients in the melt pool. The maximum velocity of melt flow is close to 2.5 m/s. The length and depth of the melt pool are 0.235 mm and 0.0265 mm respectively.

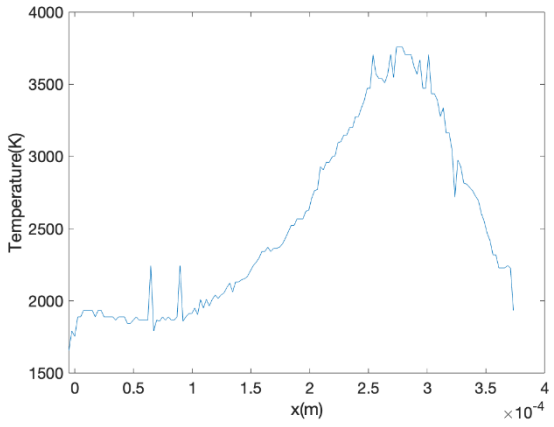
The thermal aspect of the model is validated by comparing the temperature predictions and experimental results. The velocity field cannot be validated, because velocity field was not measured in this experiment [47]. The temperature readings above 1500 K along the top edge of the 2D melt pool domain are compared with the temperature readings along the centreline in Figure 37. The comparison is shown in Figure 39. The temperature distributions from experiments and the forward modelling are similar. The profile is measured from the tail of the melt pool towards the head, the temperature gradually increases to above the melting point of Ti6Al4V around 1900 K, and then keeps almost constant for a short distance because latent heat is generated during the phase change. The temperature further increases as it is closer to the laser centre. The highest temperature is about 3600 K. The location of the maximum temperature is around $x = 0.0003$ m from where the melt pool starts. The major discrepancy between the forward modelling prediction and experimental measurement is the temperature gradient before and after the peak temperature. The temperature gradient from the forward modelling is higher than that from experiments. This is likely caused by the vaporization of powders near the laser centre, which was not considered in the physical model. There are also noises in the experimental results, which could come from the vaporization of powder particles and high-temperature particles flying out of the laser's path.

(a) Temperature field

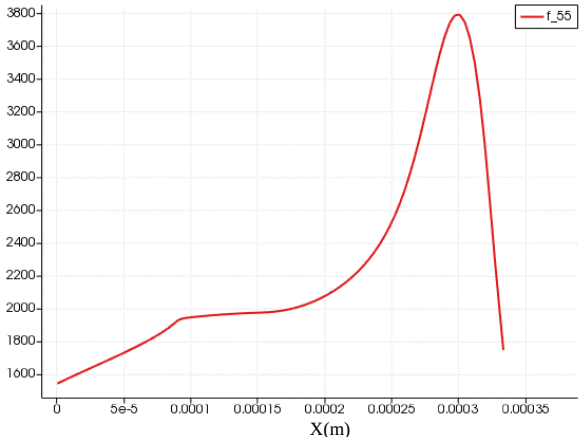


(b) Velocity field

Figure 38. Temperature and velocity fields from forward modelling



(a) Temperature from experiments



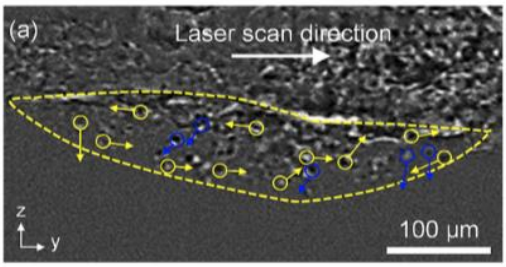
(b) Temperature from forward modelling

Figure 39. Comparison of temperature distributions

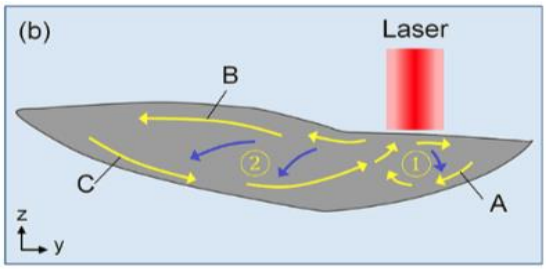
The fluid aspect of the forward model is validated by comparing its predicted melt flow velocities with the experimental results from the work of Guo et al. [58], where AlSi10Mg was used in the SLM process. Therefore, a different set of model parameters need to be used to match

the material properties and process parameters. The melt pool dynamics in the experiment was measured by a high-speed high-resolution x-ray imaging system as shown in Figure 40(a). Different flow patterns in the melt pool are illustrated in Figure 40(b). Flow type A is the flow at the front portion of melt pool and in the backward direction at the bottom. Flow type B is the flow backward near the melt pool surface behind the laser. Flow type C is the forward flow along the melt pool bottom in the rear portion of melt pool. The velocities at some locations in the melt pool were estimated by tracing the movement of tungsten micro-particles that were uniformly mixed with AlSi10Mg powders. The laser power and scanning speed are 260 W and 0.6 m/s respectively. The effective beam radius is 40 μm. The temperature and velocity fields obtained from the forward modelling are shown in Figure 41. The maximum temperature is close to 2100 K. The maximum velocity is close to 4.8 m/s. The length and depth of the melt pool are 0.445 mm and 0.091 mm respectively. Different flow patterns can be similarly seen in Figure 41(c) and match with the experimental observations in Figure 40(b). The comparisons of the maximum, average and standard deviation of velocities for the three flow types between the model prediction and experimental results are shown in Table 5. The velocity field obtained from the forward modelling agrees with the experimental results.

Here the temperature and velocity fields obtained from the physical model are validated separately, because the simultaneous measurements of the two are very challenging due to the limitations of instrument. By comparing the two separate experimental results with different materials and process parameters, we have seen the robustness of the physical model. During the calibration and validation, model parameters such as the mesh size and solver are kept to be the same for both cases.

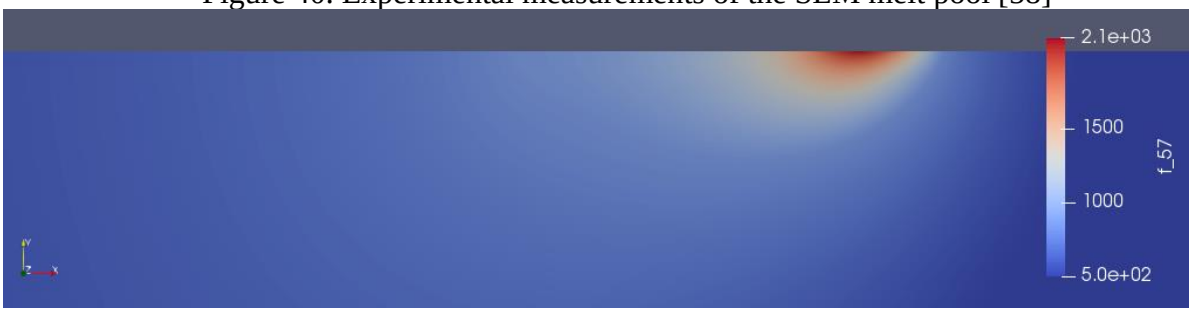


(a) X-ray image of the SLM melt pool

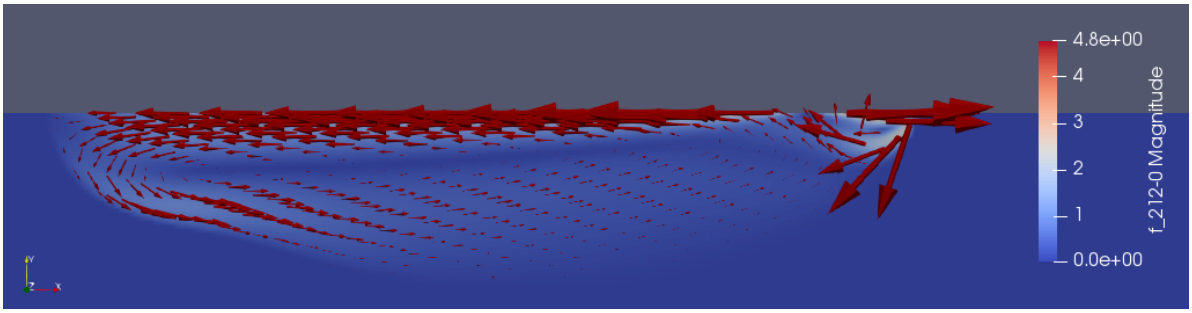


(b) Flow pattern in the SLM melt pool

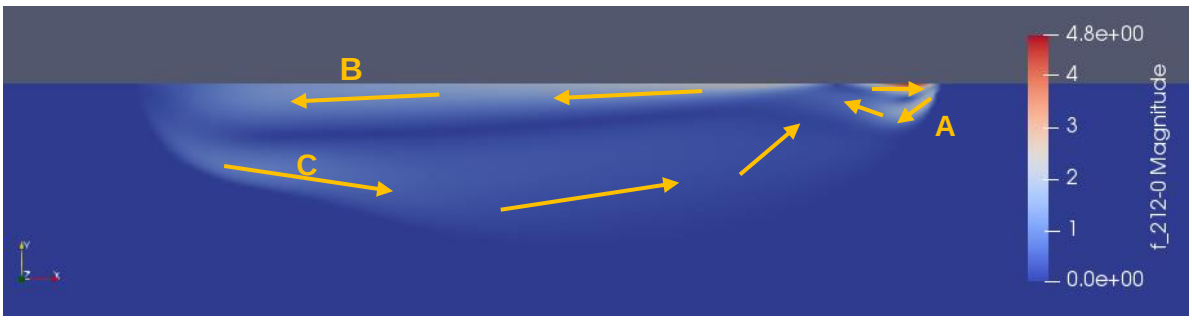
Figure 40. Experimental measurements of the SLM melt pool [58]



(a) Temperature field



(b) Velocity field



(c) Patterns of velocity field

Figure 41. Temperature and velocity fields from forward modelling for velocity field validation

Table 5. Velocity for different flow types

Flow type	Max velocity (m/s)		Average velocity (m/s)		Standard deviation of velocity (m/s)	
	Experimental results [58]	Model Predictions	Experimental results [58]	Model Predictions	Experimental results [58]	Model Predictions
A	2.84	2.53	1.78	1.65	0.79	0.62
B	1.38	1.61	0.95	1.02	0.30	0.26
C	1.05	1.12	0.52	0.61	0.23	0.29

5.3 PBCS framework for Melt Pool Monitoring

The basic idea of PBCS for melt pool temperature and melt flow measurement is to first recover the heat load vector in Eq. (5.28) or fluid flow force vector in Eq. (5.27) in the thermofluid model developed in Section 5.2 from some actual measurements. Then the forward model is applied to predict the complete temperature and velocity fields. Theoretically, if all necessary model parameters such as effective laser power intensity on the surface, radiation, convection, and Marangoni force are readily available, the temperature and velocity fields can be directly predicted from the physical model of melt pool. However, the accurate values of these parameters are usually unavailable and they cannot be measured directly during the manufacturing process. Therefore, the method of indirect measurement and reconstruction as in PBCS is very necessary. In this section, we describe the two PBCS formulations of reconstruction based on temperature measurement and velocity measurement.

5.3.1 PBCS based on measurements of temperature

The PBCS mechanism is established based on the finite-element discretization of the multi-physics model. The transient problem in Eq. (5.28) can be approximated by the backward difference and reformulated as

$$\boldsymbol{T}^{(n+1)} - \boldsymbol{\alpha}\left(\frac{\mathbf{M}}{\Delta t} - \mathbf{D}\left(\boldsymbol{u}^{(n)}\right)\right)\boldsymbol{T}^{(n)} = \boldsymbol{\alpha}\boldsymbol{L} \tag{5.30}$$

where $\boldsymbol{\alpha} = (\mathbf{M}/\Delta t + \mathbf{K})^{-1}$. Δt is the time step. $\boldsymbol{u}^{(n)}$ and $\boldsymbol{T}^{(n)}$ are velocity and temperature fields from the *n*-th iteration. A binary measurement matrix $\boldsymbol{\Phi_T}$ is constructed to store the indices of nodes as the locations of the limited temperature measurements. Given a total of N nodes, the measurement matrix $\boldsymbol{\Phi_T}$ has the size of $M \times N$, where M is the number of measurements and $M \ll N$.

The load vector $\boldsymbol{L}$ that needs to be recovered is a sparse vector in the physical model. Sparsity is important for the efficiency and accuracy of PBCS recovery. The recovery is to find the load vector $\boldsymbol{L}$ such that the difference between the experimental measurements $\boldsymbol{T}^*$ and the corresponding observations $\widetilde{\boldsymbol{T}}^{(n+1)} = \boldsymbol{\Phi_T}\boldsymbol{T}^{(n+1)}$ from the model is minimized by solving

$$\min\left\|\boldsymbol{T}^* - \widetilde{\boldsymbol{T}}^{(n+1)}\right\|_2 \tag{5.31}$$

subject to the constraints in Eq.(5.30) and

$$\|\boldsymbol{L}\|_0 < s_l \tag{5.32}$$

The recovery is based on the OMP algorithm in Table 1. With the recovered load vector $\boldsymbol{L}$, the temperature field can be reconstructed by solving the forward model in Eq. (5.28). The reconstructed temperature field is then used to predict boundary conditions for the fluid flow model such as the buoyancy force in Eq. (5.14) and Marangoni force in Eq. (5.18). The complete velocity field is then reconstructed by solving the forward problem based on Eqs. (5.26) and (5.27).

The PBCS algorithm for the multi-physics system is summarized in Table 6. In the PBCS algorithm, computing $\boldsymbol{\alpha}$ directly by inverting the square matrix can be computationally expensive for high-fidelity models with a large number of nodes. Here $\boldsymbol{\Phi_T}\boldsymbol{\alpha}$ is computed together. Computing $\boldsymbol{\Phi_T}\boldsymbol{\alpha}$ only needs M rows of $\boldsymbol{\alpha}$ since there are only a limited number of measurements.

Denote $\boldsymbol{\beta} = \boldsymbol{\alpha}^{-1}$ and $\boldsymbol{\alpha}\boldsymbol{\beta} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Calculating $\boldsymbol{\beta}$ is equivalent to solve $\boldsymbol{\alpha}\boldsymbol{\beta}_i = \mathbf{e}_i, i = 1, \dots, N$, where $\boldsymbol{\beta}_i$ is the i^{th} column of $\boldsymbol{\beta}$ and $\mathbf{e}_i$ is the i^{th} column of $\mathbf{I}$. With the notation $\boldsymbol{\omega} = \boldsymbol{\Phi}\boldsymbol{\alpha}$, $\boldsymbol{\omega}$ can be found by solving $[\mathbf{M}/\Delta t + \mathbf{G}]^{\mathbf{T}}\boldsymbol{\omega}^{\mathbf{T}} = \boldsymbol{\Phi}_{\boldsymbol{T}}{}^{\mathbf{T}}$ in the steady state case. Since matrices $[\mathbf{M}/\Delta t + \mathbf{G}]^{\mathbf{T}}$ and $\boldsymbol{\Phi}_{\boldsymbol{T}}{}^{\mathbf{T}}$ are sparse, the computation of $\boldsymbol{\omega}$ can be much more efficient when $M \ll N$.

Table 6. PBCS algorithm for the multi-physics system based on temperature measurements

Input:	measurements $\boldsymbol{T}^*$; measurement matrix $\boldsymbol{\Phi}_{\boldsymbol{T}}$; model matrices $\mathbf{M_u}, \mathbf{C}, \mathbf{K_u}, \mathbf{M}, \mathbf{K}, \mathbf{D}$ and $\mathbf{Q}$; Time step Δt and total time $\hat{t}$ for the transient model.
Output:	Estimated load vector $\boldsymbol{L}$. Complete velocity and temperature field.

Procedure:	Initialize velocity $\boldsymbol{v}^{(0)}$, temperature $\boldsymbol{T}^{(0)}$, and time $t = 0$.
	WHILE $t < \hat{t}$ DO
	1. Recover the load vector $\boldsymbol{L}$ by solving Eq.(5.31) with the constrained OMP algorithm
	2. Reconstruct temperature field by solving Eq. (5.28) with recovered $\boldsymbol{L}$
	3. Estimate boundary conditions for the fluid flow model with reconstructed temperature field
	4. Reconstruct velocity field by solving Eqs. (5.26) and (5.27)
	5. $t = t + \Delta t$
	END WHILE

5.3.2 PBCS based on measurements of velocity

As shown in Section 5.3.1, if only temperature measurements are available, the heat load vector $\boldsymbol{L}$ can be recovered. The boundary conditions for the fluid flow are then estimated with reconstructed temperature field. If some velocities can be measured directly, the load vector $\boldsymbol{F}$ in Eq. (5.27) including boundary conditions for the fluid flow model can be recovered with the

velocity measurements. The conservation of momentum in Eq. (5.27) can be approximated by the backward difference and reformulated as

$$\boldsymbol{u}^{(n+1)} - \boldsymbol{\eta}\left(\frac{\mathbf{M_u}}{\Delta t}\boldsymbol{u}^{(n)} - \mathbf{Q}\boldsymbol{p}^{(n)}\right) = \boldsymbol{\eta}\boldsymbol{F_u} \tag{5.33}$$

where $\boldsymbol{\eta} = \left(\mathbf{M_u}/\Delta t + \mathbf{C}\left(\boldsymbol{u}^{(n)}\right) + \mathbf{K_u}\right)^{-1}$. $\boldsymbol{u}^{(n)}$ and $\boldsymbol{p}^{(n)}$ are velocity and pressure fields from the *n*-th iteration. The measurement matrix $\boldsymbol{\Phi_u}$ is constructed to store the indices of nodes or locations of the limited velocity measurements.

The PBCS recovery is to find the load vector $\boldsymbol{F_u}$ such that the difference between the experimental measurements $\boldsymbol{u}^*$ and the corresponding observations $\widetilde{\boldsymbol{u}}^{(n+1)} = \boldsymbol{\Phi_u}\boldsymbol{u}^{(n+1)}$ from the model is minimized by solving

$$\min\left\|\boldsymbol{u}^* - \widetilde{\boldsymbol{u}}^{(n+1)}\right\|_2 \tag{5.34}$$

subject to the constraints in Eq.(5.33) and

$$\|\boldsymbol{F_u}\|_0 < \boldsymbol{s}_l \tag{5.35}$$

Similarly, the recovery can be done based on the constrained OMP algorithm. With the recovered load vector $\boldsymbol{F_u}$, the complete velocity and temperature fields can be reconstructed with Eqs. (5.26-5.28).

5.4 Demonstration of PBCS framework

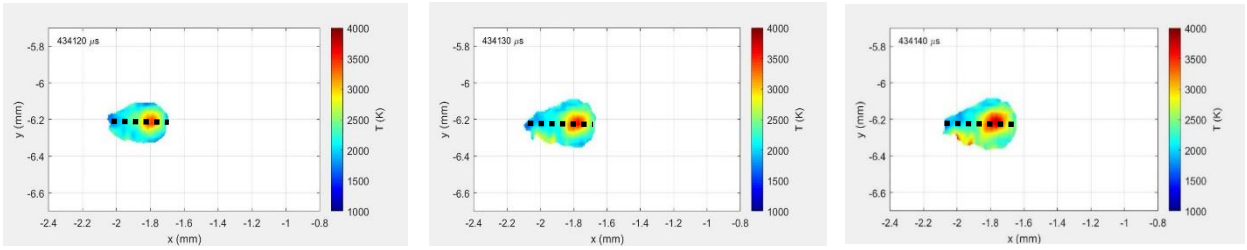
The proposed PBCS framework is used to monitor the temperature and velocity fields in the SLM process. The experimental results of temperature distribution obtained from Hooper's work [47] are used to recover the heat load vector as in Eqs. (5.31-5.32). Then the complete temperature and velocity fields are reconstructed. The formulation is described in Section 5.3.1. The PBCS formulation with direct velocity measurements in Section 5.3.2 is not used here because there was

no measurement of velocity in the experiment [47]. The metallic material in the experiment was Ti6Al4V.

To start the recovery process, the initial guesses of temperature field $\boldsymbol{T}^{(0)}$ and velocity field $\boldsymbol{u}^{(0)}$ at time t_0 are required. They are obtained as follows. With the initial guess of process parameters for the forward modelling such as the heat flux from laser source, convection, and radiation, the temperature field $\boldsymbol{T}^{(F)}$ and velocity field $\boldsymbol{u}^{(F)}$ at a time step prior to t_0 are estimated by solving Eqs. (5.26-5.28), as shown in Figure 43(a) and (b). Experimental results from the centreline of the melt pool shown in Figure 42(a) are mapped to the top edge of the 2D model in Figure 43(c). With these measurements at time t_0 and the measurement matrix $\boldsymbol{\Phi_T}$, which indicates the nodal indices of measurements on the top edge of the model, and initial guess of the temperature field $\boldsymbol{T}^{(0)}$, Eq.(5.30) can be rewritten as

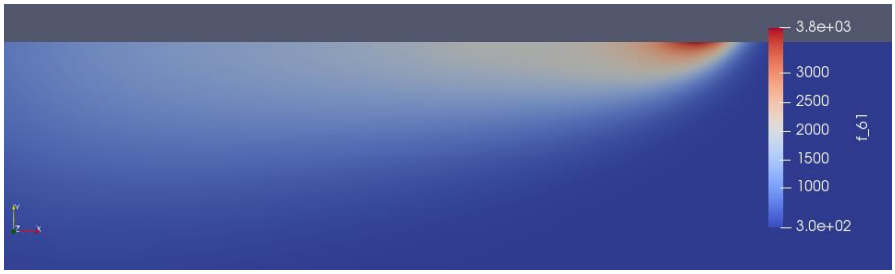
$$\boldsymbol{T}^{(0)} - \boldsymbol{\alpha}\left(\frac{\mathbf{N}}{\Delta t} - \mathbf{D}\left(\boldsymbol{u}^{(F)}\right)\right)\boldsymbol{T}^{(F)} = \boldsymbol{\alpha L} \tag{5.36}$$

The heat load vector $\boldsymbol{L}$ can be recovered based on Eqs. (5.31), (5.32) and (5.36). The initial temperature field $\boldsymbol{T}^{(0)}$ and velocity field $\boldsymbol{u}^{(0)}$ shown in Figure 44 can be reconstructed by solving Eqs. (5.26-5.28) with recovered $\boldsymbol{L}$.

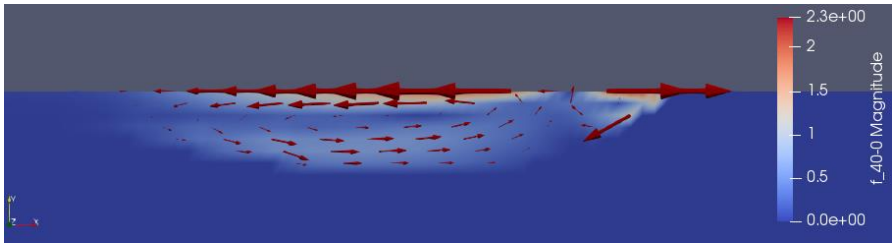


(a) Measured temperature at t_0 (b) Measured temperature at t_1 (c) Measured temperature at t_2

Figure 42. Experimentally measured temperatures at time t_0, t_1 and t_2



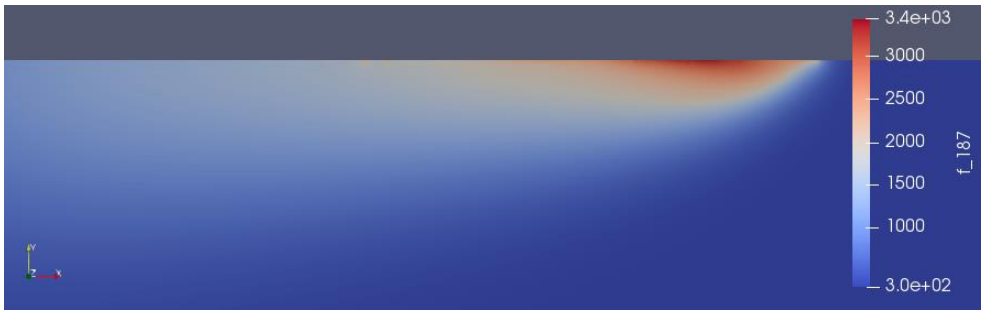
(a) Initial guess of temperature field from forward modelling



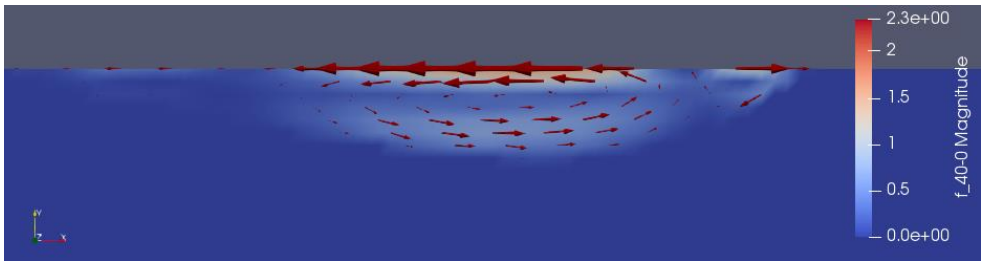
(b) initial guess of velocity field from forward modelling

(c) Experimental temperature results at t_0

Figure 43. Required inputs to reconstruct initial temperature and velocity fields



(a) Initial temperature field



(b) Initial velocity field

Figure 44. Reconstructed temperature and velocity fields at t_0

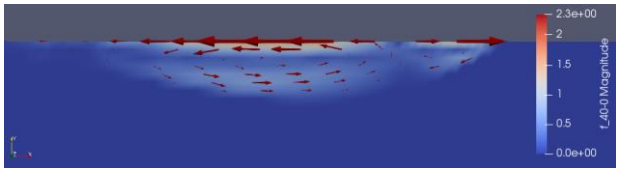
The available experimental temperature measurements at t_1 and t_2 are shown in Figure 42(b) and (c). The time step between two successive frames is 10 μs. The temperatures at the centreline of the melt pool at t_2 are extracted for the recovery of the heat load vector $\boldsymbol{L}$. 108 measured nodal temperatures at the top edge are used to reconstruct a total of 32481 nodal temperatures in the 2D domain. Therefore, the compression ratio is 32481/108=300.75. With the recovered $\boldsymbol{L}$, the complete temperature and velocity fields at t_1 and t_2 as well as other intermediate time steps $t_{0.5}$ and $t_{1.5}$ can be reconstructed, as shown in Figure 45.

For quantitative validation, the PBCS reconstructed temperature readings above 1500 K indicated as the dash line in Figure 46 (a) is compared with the available experimental measurements along the centreline of melt pool top surface at time t_1. The relative PBCS reconstruction errors along the centreline are shown in Figure 46 (b). The maximum reconstruction error is 16.47%, the average error is 3.34%, and standard deviation of errors is 3.57%. The largest

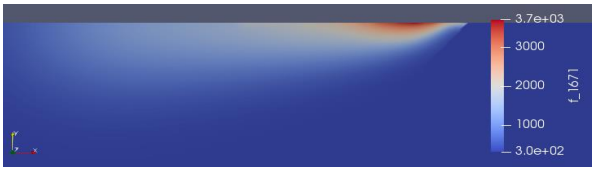
relative errors are found at both ends of the melt pool, where the temperatures are also the lowest in the melt pool. Similar to most sensors, the measurements toward the limits of measurement range tend to have the largest errors. If the errors at these two ends of the melt pool are excluded, the maximum, average, and standard deviation of errors are reduced to 6.32%, 2.02%, and 1.49% respectively. Because of the lack of simultaneous measurements of temperature and velocity, the reconstructed velocity field cannot be quantitatively validated. Qualitatively the velocity pattern matches the one observed in the SLM process of AlSi10Mg in Section 5.2.3.

In this example, the computational time for recovery is 4.87 s and reconstruction is 4.77 s for one time step on a single-CPU desktop computer.

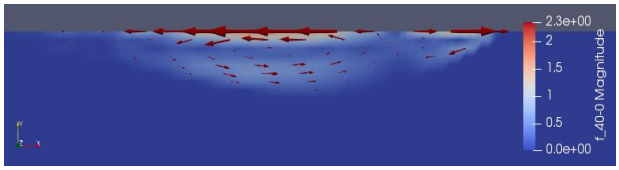
(a) Reconstructed temperature field at $t_{0.5}$



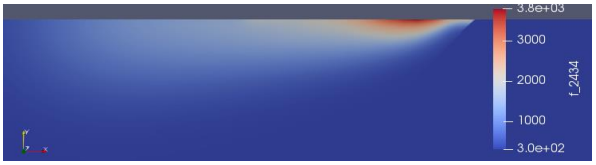
(b) Reconstructed velocity field at $t_{0.5}$



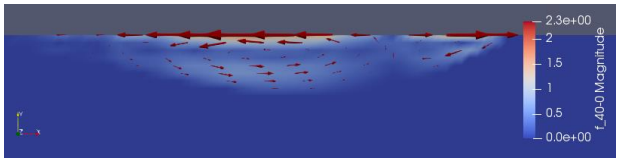
(c) Reconstructed temperature field at t_1



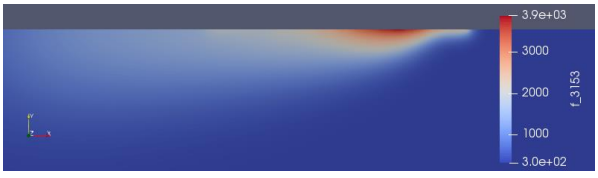
(d) Reconstructed velocity field at t_1



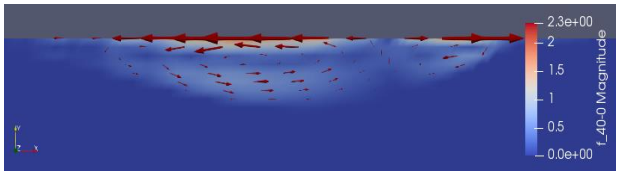
(e) Reconstructed temperature field at $t_{1.5}$



(f) Reconstructed velocity field at $t_{1.5}$

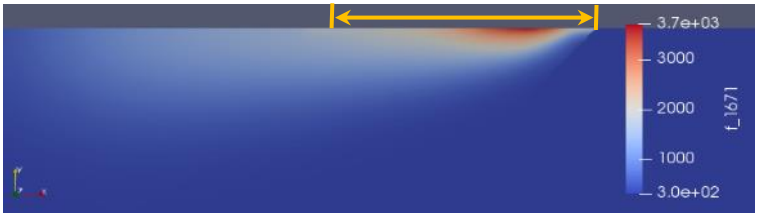


(g) Reconstructed temperature field at t_2

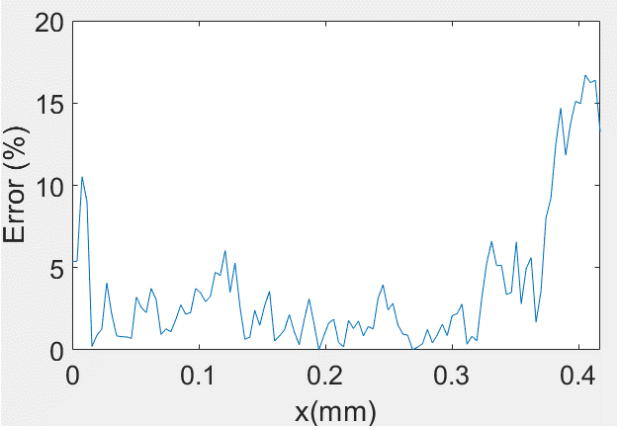


(h) Reconstructed velocity field at t_2

Figure 45. Reconstructed temperature and velocity fields



(a) Region with temperature higher than 1500 K for comparison



(b) Reconstruction errors

Figure 46. Reconstruction errors of temperatures in the melt pool at t_1

5.5 Conclusion

The PBCS mechanism is developed to simultaneously monitor the temperature and velocity fields in the SLM process. The PBCS mechanism is a new hybrid physics-based data-driven approach to perform sensing, where multi-physics modelling is utilized to reconstruct what we cannot measure directly. PBCS helps overcome the limitations of conventional sensing due to sensor accessibility, unobservable quantities of interest, and temporal and spatial resolutions. For instance, the temperature distribution and velocity field reconstructed from PBCS have much higher resolutions than those directly measured by conventional sensors. 3D temperature and melt flow can also be obtained from a few measured temperature readings on the top surface. Therefore, PBCS provides a strategy to monitor complex processes such as melt flow when cost-effective sensors are unavailable.

The main PBCS procedure is to build correlations between measurable quantities, typically

on the boundaries of domains, and the unobservable internal distributions, as well as inherent correlations between different physical quantities such as between temperature distribution and velocity field. Prior knowledge embodied as physical models is necessary in building the correlations. It is not easy to quantitatively validate multi-physics models in PBCS because of the limitation in available experimental techniques to simultaneously measure multiple quantities. The predictions of different quantities usually need to be validated separately with multiple experimental datasets, as demonstrated in this work. We showed that by using real-time measurements to recover the boundary conditions of transient models such as the load vector of the heat transfer model, the complete temperature distributions and velocity fields can be reconstructed. In complex manufacturing processes, the boundary conditions associated with the physical models are dynamically changing, and the direct measurements of them are difficult. Therefore, the inverse method that recovers the boundary conditions from some temperature measurements is more attractive than direct measurement. The PBCS framework has the advantages of obtaining more complete information from limited measurable data, which makes it a cost-effective and efficient approach for process monitoring.

The major challenge of PBCS for in-situ process monitoring is the computational cost associated with recovery and reconstruction, where physical models are involved. In the thermofluid model, the computation of 3D fluid flows is particularly expensive. In this Chapter, 2D model of melt pool was used to reduce the computational cost. The PBCS based on 2D model can be easily extended to 3D model. However, the size of matrices used in recovery and reconstruction can increase quickly, especially when the mesh density needs to remain high to ensure the convergence of fluid dynamics modelling. Another challenge of PBCS is the accuracy of the physical models. Simplification and approximation are always involved with modelling. For

instance, in the thermofluid model developed, detailed physics such as vaporization due to high laser power and the effects of the melt pool dynamics on surface roughness are not considered. Surface roughness of the melt pool can affect capillary and Marangoni forces which depend on the surface curvature and surface tension. Systematic errors will be introduced because of the models. Trade-offs between efficiency and accuracy need to be made. If more physics is included in the physical model of PBCS, reconstructed temperature distribution and velocity field can be more accurate, but at a higher cost of computation. The number of available measurements also affects PBCS performance. With more measurements, reconstructed results are more accurate. For example, if velocities on the top surface of melt pool can be measured, the load vector of the fluid flow model can also be recovered. The accuracy of the PBCS reconstructed velocity field can be further improved.

In future work, 3D physical models will be integrated in the PBCS framework to reconstruct 3D temperature and velocity fields in metal AM processes. The efficiency of PBCS needs to be improved for the application of in-situ monitoring. The obvious way is to accelerate the computation is using parallel computers. The developed PBCS formulation and models can be easily parallelized. The multi-physics model will be further developed to include more physics such as the vaporization in the depression zone of the melt pool when laser power is high, as well as the shape change of metal powders during the melting process. This extension will improve the fidelity of models and further generalize the PBCS framework for more applications.

CHAPTER 6. Physics-Constrained Dictionary Learning for One-Dimensional Signal Compression

6.1 Introduction

The existing techniques to monitor AM processes has been reviewed in Section 2.2. In addition to monitor the temperature distribution in AM processes, machine conditions and build qualities can also be monitored with other techniques such as acoustic emission (AE) [33, 34, 35]. Abnormal conditions in manufacturing processes can be detected based on the patterns of 1D signals collected by AE sensor. However, as AE signals are continuously collected, transmitted, and stored, the available bandwidth in communication channels for transmission of a large amount of data is limited. Therefore, reducing the amount of data in communication and storage without sacrificing the amount of information collection is necessary.

The CS introduced in Section 2.1 was developed as a new sampling approach which requires less data storage and the original signal can be reconstructed. The existing dictionary learning approaches described in Section 2.4 was used to improve the sparsity level of signals. However, they cannot be used to determine the time stamps of data samples to be measured and stored for 1D signals, because the measurement matrix in CS is not designed explicitly. For instance, to select data points for storage and communication from all collected signals, the measurement matrix needs to be optimized to determine the total number of stored and transmitted measurements. For the measurement matrix, there should be only one non-zero entry in each row of the measurement matrix and other entries are zeros. The index of non-zero entry in each row indicates the time stamps of sampling or when to store and transmit data. Therefore, the optimization of the measurement matrix and the basis matrix in CS individually provides more physical meanings of the optimized dictionary. Furthermore, the columns of the trained dictionary in traditional

dictionary learning are not always orthogonal, which affects the CS performance. A well-designed measurement matrix can also improve the orthogonality of the columns in the trained dictionary.

A physics-constrained dictionary learning scheme is proposed to reduce the amount of data in storage and communication. From all collected data points, only a few of them are stored and transmitted. The original signal can be reconstructed from the compressed data with CS. The actual storage space and communication cost are determined by the optimized measurement matrix, and the signal can achieve a high sparsity level with respect to the optimized basis matrix. Some physical constraints such as the data storage space, the number of measurements, sensor accessibility, and the energy consumption of data collection can be considered in the learning process to optimize the basis and measurement matrices. Here, the minimum sampling interval between compressed data points is used as the physical constraint to demonstrate the new physics-constrained dictionary learning approach for 1D signals, which can reduce the redundancy for the storage and communication of temporally correlated data.

The proposed physics-constrained dictionary learning method is described in Section 6.2. The demonstration of its application to compress the roller bearing vibration signal for the storage and communication, and experimental results are given in Section 6.3.

6.2 Methodology

The proposed physics-constrained dictionary learning scheme is to optimize the measurement matrix $\boldsymbol{\Phi}$ and the basis matrix $\boldsymbol{\Psi}$ simultaneously under the physical constraints related to the time stamps for sampling. It is formulated as

$$\min_{\Phi,\Psi,\boldsymbol{\Upsilon}} \ \alpha\|\mathbf{S}-\boldsymbol{\Psi}\boldsymbol{\Upsilon}\|_F^2+\|\boldsymbol{\Phi}\mathbf{S}-\boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\Upsilon}\|_F^2 \tag{6.1}$$

$$\text{subject to } \boldsymbol{\Phi}=f(\boldsymbol{\Psi}) \tag{6.2}$$

$$\|\boldsymbol{\gamma}_i\|_{\mathbf{0}} \leq \ s_l, \ \ \forall i \tag{6.3}$$

$$I_{ij}(\boldsymbol{\Phi}) \geq r, \quad \forall i, j \tag{6.4}$$

where F denotes the Frobenius norm, $\mathbf{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2 \ldots \boldsymbol{s}_P] \in \mathbb{R}^{N \times P}$ contains P sets of training signals and each set of signals has the length of N. $\boldsymbol{\Psi} \in \mathbb{R}^{N \times W}$ is the basis matrix with $W \ll P$ and $W > N$. $\boldsymbol{\Upsilon} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 \ldots \boldsymbol{\gamma}_P] \in \mathbb{R}^{W \times P}$ contains the sparse coefficients that represent the training signals in $\mathbf{S}$ with respect to the basis matrix. A Lagrange multiplier α is applied to combine the objectives of recovery accuracy and measurement accuracy. A small value of α such as 0.01 is used in practice because a relatively larger control weight of the error term $\|\boldsymbol{\Phi}\mathbf{S} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\Upsilon}\|_F^2$ is necessary to design the measurement matrix to minimize the reconstruction error. The constraint in Eq.(6.2) indicates the training sequence, where basis matrix $\boldsymbol{\Psi}$ is updated before measurement matrix $\boldsymbol{\Phi}$. With the fixed basis matrix $\boldsymbol{\Psi}$, measurement matrix $\boldsymbol{\Phi}$ can be optimized based on $f(\boldsymbol{\Psi})$. The constraint in Eq.(6.3) is the upper limit of the sparsity level, where $\boldsymbol{\gamma}_i$ is the i-th column of coefficient matrix, and s_l is the target number of non-zero values in the sparse vectors of coefficients. The constraint in Eq.(6.4) shows the physical limitations of sampling, which is the lower limit of the time interval I_{ij} between the i-th and j-th stored or transmitted measurements, for instance, between any two consecutive measurements. If the time interval between stored or transmitted measurements is too small, more redundant information is collected because of large similarities between temporally correlated measurements. Other physical constraints can be added similarly.

The physics-constrained dictionary learning problem is solved to optimize the measurement and basis matrices by two stages iteratively as shown in Figure 47. It starts with an initial guess of the basis matrix which can be some known transformation matrices such as discrete cosine transformation and wavelet transformation. In the first stage, with the basis matrix fixed, the measurement matrix is optimized by determining suitable measurements to be stored and

transmitted from originally collected signals under the constraint that there is only one non-zero entry in each row to determine the time stamp of the measurement. This can be solved based on algorithms such as the FrameSense [120]. In the second stage, with the measurement matrix fixed, the basis matrix is then optimized. This can be done with dictionary learning algorithms such as the K-SVD [84]. The above two optimization steps are repeated until both the optimal measurement and basis matrices converge without further improvement. Physical constraints such as the total number of samples and the minimum sampling interval can be incorporated.

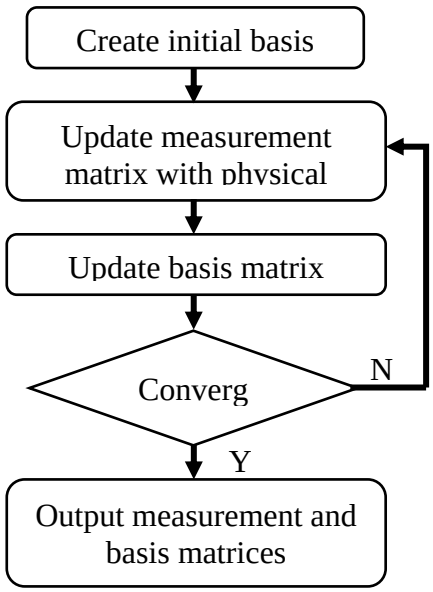


Figure 47. Two-stage optimization scheme.

6.2.1 Stage one optimization

At the stage one, the basis matrix $\mathbf{\Psi}$ is fixed, and the measurement matrix $\mathbf{\Phi}$ is optimized. Instead of directly searching for the optimal time stamps of stored and transmitted measurements which is often NP-hard if the number of measurements is large, the near-optimal time stamps of sampling can be obtained based on the greedy algorithm FrameSense in Table 7. Given all available time stamps $\mathcal{N} = \{1, \dots, N\}$, an unsuitable set of time stamps $\mathcal{T}$ can be iteratively

identified as the index of the row in the basis matrix $\mathbf{\Psi}$ by solving

$$\max_{\mathcal{T}} \mathrm{F}(\mathcal{T}) = \mathrm{H}(\mathbf{\Psi}) - \mathrm{H}(\mathbf{\Psi}_{\mathcal{N}\backslash\mathcal{T}}) \tag{6.5}$$

where $\mathrm{H}(\mathbf{\Psi})$ is the so-called frame potential and represented as

$$\mathrm{H}(\mathbf{\Psi}) = \sum_{i=1}^{N} |\lambda_i|^2 \tag{6.6}$$

where λ_i is the i-th largest eigenvalue of $\mathbf{\Psi}^*\mathbf{\Psi}$ and $\mathbf{\Psi}^*$ is the conjugate transpose of $\mathbf{\Psi}$. $\mathbf{\Psi}_{\mathcal{N}} = \mathbf{\Psi}$ all time stamps of measurements are used. $\mathbf{\Psi}_{\mathcal{N}\backslash\mathcal{T}}$ is a sub-matrix of $\mathbf{\Psi}_{\mathcal{N}}$ with rows corresponding to indices with the unsuitable ones excluded. After determining the unsuitable time stamps $\mathcal{T}$, the new available time stamps are updated as $\mathcal{N}\backslash\mathcal{T}$. If M measurements are desirable, the time stamps of M measurements are optimized by excluding $(N - M)$ unsuitable time stamps iteratively. Eventually the time stamps of the desirable measurements can be identified in the optimized $M \times N$ measurement matrix in a form of Eq.(3.12), where the column index of the value of 1 in each row indicates the time of each stored and transmitted measurement.

Here, the original FrameSense algorithm is modified to check if the additional physical constraints in Eq.(6.4) are satisfied. If the time interval between any of two stored and transmitted measurements is less than the threshold value r, one of the measurements in the pair is eliminated. For 1D signals, r indicates the minimum time interval between two adjacent samples.

Table 7. The constrained FrameSense algorithm for 1D signals

1. Initialize time stamps of stored and transmitted measurements $\mathcal{L}$, all available time stamps $\mathcal{N}$, and desired number of stored and transmitted measurements m_t
2. Determine the first two removed rows in $\mathbf{\Psi}$ by solving $\mathcal{T} = \text{argmax}_{i,j\in\mathcal{N}} \left| < \boldsymbol{\psi}_i, \boldsymbol{\psi}_j > \right|^2$ and update remaining time stamps $\mathcal{L} = \mathcal{N} \backslash \mathcal{T}$ by excluding $\mathcal{T}$
3. WHILE the length of $\mathcal{L} < m_t$ DO
   Find the i^*-th row in $\mathbf{\Psi}$ to eliminate by solving $i^* = \text{argmax}_{i\in\mathcal{L}} \text{F}(\mathcal{T} \cup \{i\})$, where $\text{F}(\mathcal{T} \cup \{i\})$ is the function in Eq.(6.5)
   Update unsuitable time stamps of stored and transmitted measurements as $\mathcal{T} = \mathcal{T} \cup \{i^*\}$
   Update available time stamps of stored and transmitted measurements, $\mathcal{L} = \mathcal{L} \backslash \{i^*\}$
   END WHILE
4. FOR i = 1 to the length of $\mathcal{L}$
   FOR j = 1 to length of $\mathcal{L}$
   If $I_{ij}(\mathbf{\Phi}) = \left| t_i - t_j \right| \leq r$, where t_i and t_j are time stamps for i-th and j-th data, $\mathcal{L} = \mathcal{L} \backslash \{j\}$.
   END FOR
   END FOR
5. Generate measurement matrix $\mathbf{\Phi}$ with optimized time stamps $\mathcal{L}$

6.2.2 Stage two optimization

A simplified form of the objective function in Eq.(6.1) can be written as

$$\min_{\Phi,\Psi,\Upsilon} = \left\| \begin{pmatrix} \alpha\mathbf{S} \\ \mathbf{\Phi S} \end{pmatrix} - \begin{pmatrix} \alpha\mathbf{I} \\ \mathbf{\Phi} \end{pmatrix} \mathbf{\Psi\Upsilon} \right\|_F^2 \tag{6.7}$$

At the stage two, $\mathbf{\Psi}$ and $\mathbf{\Upsilon}$ are optimized with the fixed measurement matrix $\mathbf{\Phi}$ obtained in the stage one subject to the constraint in Eq.(6.3). With new notations $\mathbf{X} = \begin{pmatrix} \alpha\mathbf{S} \\ \mathbf{\Phi S} \end{pmatrix}$ and $\mathbf{Z} = \begin{pmatrix} \alpha\mathbf{I} \\ \mathbf{\Phi} \end{pmatrix} \mathbf{\Psi}$, Eq.(6.7) can be rewritten, and $\mathbf{Z}$ and $\mathbf{\Upsilon}$ can be optimized by solving

$$\min_{\mathbf{Z},\boldsymbol{\Upsilon}} \|\mathbf{X} - \mathbf{Z}\boldsymbol{\Upsilon}\|_F^2 \tag{6.8}$$

$$\text{subject to } \|\boldsymbol{\gamma}_i\|_\mathbf{0} \leq s_l, \ \ \forall i \tag{6.9}$$

The optimal $\mathbf{Z}$ and $\boldsymbol{\Upsilon}$ are found iteratively. With an initialized and fixed basis matrix $\boldsymbol{\Psi}$, the coefficient matrix $\boldsymbol{\Upsilon}$ can be obtained with the OMP algorithm in solving Eqs.(6.8) and (6.9). With the obtained coefficient matrix $\boldsymbol{\Upsilon}$ fixed, one column of the basis matrix is then updated each time by solving Eq.(6.8), which is re-written as

$$\min_{\mathbf{z}_k} \left\|(\mathbf{X} - \sum_{j\neq k}^{W} \boldsymbol{z}_j \boldsymbol{\gamma}'_j) - \boldsymbol{z}_k \boldsymbol{\gamma}'_k\right\|_F^2 = \min_{\mathbf{z}_k} \|\mathbf{E}_k - \boldsymbol{z}_k \boldsymbol{\gamma}'_k\|_F^2 \tag{6.10}$$

where $\boldsymbol{z}_j$ is the *j*-th column of $\mathbf{Z}$, $\boldsymbol{\gamma}'_j$ indicates the *j*-th row of $\boldsymbol{\Upsilon}$, and $\mathbf{E}_k$ represents the errors of all sample signals except *k*-the atom. Here, $\boldsymbol{z}_k$ as the *k*-th column of $\mathbf{Z}$ is updated iteratively. In order to satisfy the sparsity constraint in Eq. (6.9), additional modification of the objective function is needed. We define

$$\omega_k = \{i | 1 \leq i \leq W, \boldsymbol{\gamma}'_k(i) \neq 0\} \tag{6.11}$$

as the set of indices where training signals $\{\boldsymbol{s}_i\}_{i=1}^{P}$ rely on atom $\boldsymbol{z}_k$. Eq. (6.10) is further written as

$$\min_{\mathbf{z}_k} \left\|\mathbf{E}_k^{\mathrm{R}} - \boldsymbol{z}_k \boldsymbol{\gamma}_k^{\mathrm{R}}\right\|_F^2 \tag{6.12}$$

where $\mathbf{E}_k^{\mathrm{R}} = \mathbf{E}_k \boldsymbol{\Omega}_k$ and $\boldsymbol{\gamma}_k^{\mathrm{R}} = \boldsymbol{\gamma}'_j \boldsymbol{\Omega}_k$. $\boldsymbol{\Omega}_k$ is a matrix of size $P \times |\omega_k|$ with ones on the $(\omega_k(i), i)$ entries and zero elsewhere. Following the K-SVD algorithm, $\mathbf{E}_k^{\mathrm{R}}$ can be decomposed to $\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^{\mathrm{T}}$ with the singular value decomposition (SVD). Then updated $\boldsymbol{z}_k$ is the first column of $\mathbf{U}$ and the coefficient vector $\boldsymbol{\gamma}_k^{\mathrm{R}}$ is the first column of $\mathbf{V}$ multiplied by $\boldsymbol{\Delta}(1,1)$. After each column of $\mathbf{Z}$ is updated, basis matrix $\boldsymbol{\Psi}$ can be obtained with the pseudo-inverse as

$$\boldsymbol{\Psi} = (\alpha^2 \mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi})^{-1} [\, \alpha \mathbf{I} \quad \boldsymbol{\Phi}^{\mathrm{T}}] \mathbf{Z} \tag{6.13}$$

Here, a modified K-SVD algorithm is developed. Different from the conventional K-SVD

which requires a prefixed number of non-zero values in the sparse vectors, the modified K-SVD can adaptively determine the most appropriate sparsity level s_l in order to reduce the reconstruction error. Depending on different training datasets, initial basis matrices, and the number of available measurements, the optimal sparsity level s_l can be different. In the adaptive K-SVD algorithm listed in Table 8, $\mathbf{S}_n$ represents n randomly selected training data from $\mathbf{S}$. With optimized $\mathbf{\Phi}$ and $\mathbf{\Psi}$, coefficient vectors $\mathbf{\Upsilon}_\mathrm{n}$ can be recovered by solving $\mathbf{\Phi S}_\mathrm{n} = \mathbf{\Phi\Psi\Upsilon}_\mathrm{n}$ with OMP. The average of reconstruction errors is computed as $e_{ave} = 1/(N \times n) \cdot \sum_i^N \sum_j^n \left(([\mathbf{S}_n]_{ij} - [\mathbf{\Psi\Upsilon}_\mathrm{n}]_{ij})/[\mathbf{S}_n]_{ij} \times 100\%\right)$. s_l is determined adaptively by finding the minimum e_{ave}.

Table 8. Adaptive K-SVD algorithm

Input:	Training signals $\mathbf{S}$; initial basis matrix $\mathbf{\Psi}_0$; measurement matrix $\mathbf{\Phi}$, initial number of non-zero values s_l; sparsity adjustment step size Δs_l; weight of error α; number of selected training data used to compute the reconstruction error n; maximum number of iterations C; target training error e_t
Output:	Estimated sparse coefficients or parameters $\mathbf{\Upsilon}$; basis matrix $\mathbf{\Psi}$; desired sparsity level s_l

Procedure:	Initialize the average of reconstruction errors e_{ave}^0 and e_{ave}^1 with $e_{ave}^1 > e_{ave}^0$, and $m = 0$.
	WHILE $m < C$ and $e_{ave}^1 < e_{ave}^0$ DO
	$e_{ave}^0 = e_{ave}^1$
	WHILE $\|\mathbf{X} - \mathbf{Z\Upsilon}\|_F^2 > e_t$ DO
	Compute $\mathbf{\Upsilon}$ by solving Eqs.(6.8) and (6.9) with OMP.
	FOR k=1 to W
	Update ω_k with Eq.(6.11)
	Compute $\mathbf{E}_k$, $\mathbf{E}_k^\mathrm{R}$ and $\boldsymbol{\gamma}_k^\mathrm{R}$
	Apply SVD to $\mathbf{E}_k^\mathrm{R} = \mathbf{U\Delta V}^\mathrm{T}$,
	Update $\boldsymbol{z}_k$ as first column of $\mathbf{U}$
	Update $\boldsymbol{\gamma}_k^\mathrm{R}$ as first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(1,1)$.
	END FOR
	Update $\mathbf{\Psi}$ with Eq.(6.13).
	END WHILE
	Compute e_{ave}^1 with optimized $\mathbf{\Phi}$, $\mathbf{\Psi}$ and $\mathbf{S}_n$.
	$m = m + 1$
	$s_l = s_l + \Delta s_l$
	END WHILE

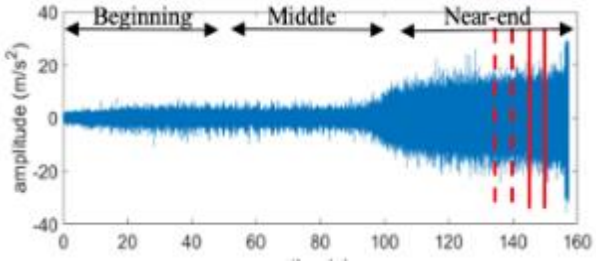
6.3 Experiments

Monitoring the health condition of machines is important for manufacturing quality control. The proposed physics-constrained dictionary learning scheme was first tested with one-dimensional signals. It was applied to compress roller bearing vibration signals to reduce the storage space and communication cost. The run-to-failure data from accelerated degradation tests of bearings were acquired by Wang *et al.*[121]. The major components of the test bed include digital force display, motor speed controller, support bearings, AC motor, hydraulic loading, tested bearing and accelerometers. Two PCB 352C33 accelerometers were used to acquire the run-to-failure data of the tested bearings. One of them was placed on the vertical axis and the other one was on the horizontal axis. The vibration signals were collected with a sampling rate of 25.6 kHz. For every minute, only the first 32768 samples (i.e. 1.26s) were recorded.

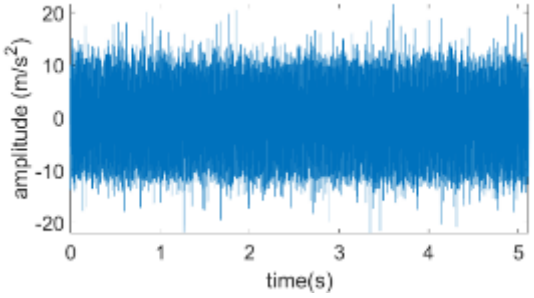
To test the proposed scheme, the vibration signals in both of horizontal and the vertical axes were used and 115000 data points were extracted from the run-to-failure data in each axis. These data points were divided into 1150 segments, and each segment contains 100 data points. These 1150 segments were used as the training dataset. For the testing dataset, 115000 data points were extracted from a different period of time. 5000 consecutive data points is then randomly selected from 115000 data points and further divided into 50 segments as the testing dataset. Training and testing datasets are extracted from different sections of the run-to-failure data such as the beginning (i.e. 0~50s), middle (i.e. 50~100s), and near end (i.e. 100~160s) of life to demonstrate the robustness of the proposed scheme. The complete and different sections of the run-to-failure data in horizontal axis are seen in Figure 48 (a). Both vibration and noise levels increase as bearings reach their later stage of lives.

Dictionary learning tends to perform better for more uniform signals in a narrower bandwidth.

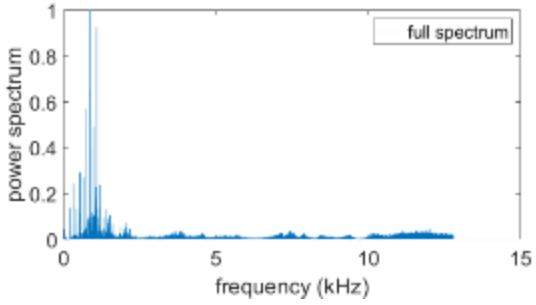
Therefore, Fourier transform is applied to decompose the original signal into signals with different bandwidths. The proposed dictionary learning scheme is applied to decomposed signals individually. The vibration signal between the vertical dash lines in Figure 48 (a) near the end of life is extracted and shown in Figure 48 (b). The extracted signal in the frequency domain after Fourier transform is shown in Figure 48 (c). The original vibration signal in Figure 48 (b) is decomposed into signals with six different bandwidths shown in Figure 49. The testing signal is extracted near the end of life data but from a different period of time, as shown in the region between two solid lines in Figure 48 (a). The extracted testing signal is shown in Figure 50 (a). The testing signal is similarly decomposed into six different bandwidths, as shown in Figure 50 (b). With the proposed physics-constrained dictionary learning scheme, one basis matrix is optimized for each decomposed training signal within the individual bandwidth. Six optimal basis matrices are obtained. Only one measurement matrix is generated, because the same time stamps are physically required during the sampling for storage and transmission for all decomposed signals. The training of the measurement matrix is based on the signal with a bandwidth of 0 ~ 2.5 kHz.



(a) The complete run-to-failure data in horizontal axis



(b) The signal extracted from the near-end of life-time



(c) The extracted signal in frequency domain

Figure 48. Extract the training signal from the complete run-to-failure data.

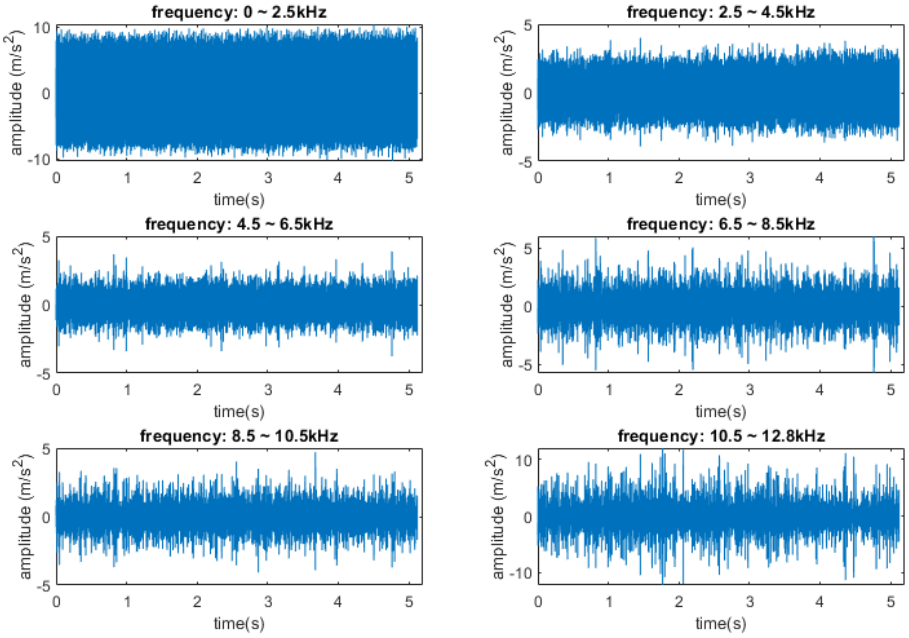
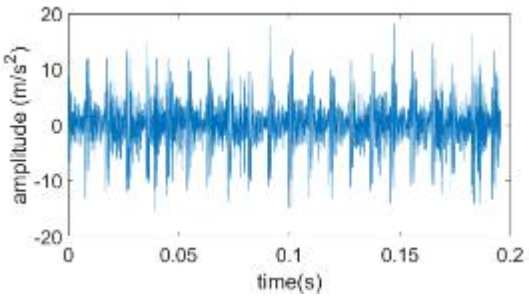
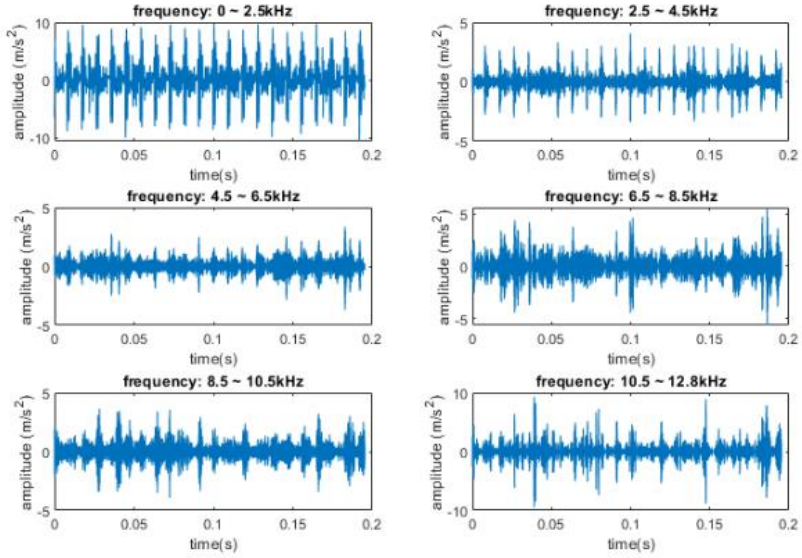


Figure 49. Decomposed training signals with different bandwidths.



(a) Original testing signal



(b) Decomposed testing signals with different bandwidths

Figure 50. Testing signal and the compositions with different bandwidths

6.3.1 Experimental results without considering the physical constraint

In the first scenario, the constraint in Eq.(6.4) that indicates the minimum sampling interval is not considered. The measurement matrix $\boldsymbol{\Phi}$ and the basis matrix $\boldsymbol{\Psi}$ are trained with the 1150 training segments. The size of the basis matrix $\boldsymbol{\Psi}$ is 100×300. The maximum number of non-zero values in each coefficient vector $\boldsymbol{\Upsilon} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 \ldots \boldsymbol{\gamma}_P]$ is determined by the modified K-SVD with adaptive s_l. Since a high sparsity level in the coefficient vectors is preferred, initial s_l is set to be 1 and Δs_l is 1. The desired s_l can be found by gradually increasing its magnitude. Among

the 100 collected data points in a segment, the number of stored and transmitted measurements was set to be 35. The time stamps for these 35 measurements are optimized with the physics-constrained dictionary learning scheme. The initial basis matrix $\boldsymbol{\Psi}$ is the discrete cosine transformation matrix. For the decomposed training and testing signals with a bandwidth of 0 ~ 2.5 kHz, the average reconstruction error for each iteration is shown in Figure 51. It is seen that the reconstruction converges after 10 iterations. In each segment, there are 100 data points and the time period is 0.0039s. The optimal 35 time stamps for samplings are selected from 100 available time stamps and marked as stars in Figure 52 (a), where unselected time stamps are marked as circles. The compression ratio is 100/35=2.86. With 35 measurements in each testing segment, all data points in all testing segments can be reconstructed with very small errors. The reconstruction errors of decomposed signals from 50 testing segments are shown in Figure 53 (a). The reconstructed signals with different bandwidths are combined and compared with the original signal in Figure 50 (a). The reconstruction errors of the combined signal are seen in Figure 53 (b). The maximum reconstruction error of the combined signal is 1.159%, the average error is 0.0024%, and the STD of errors is 0.0267%.

Sensitivity analysis is also performed with different number of stored and transmitted measurements. With 40 and 45 measurements in each segment, the maximum reconstruction errors of combined signals are 0.4256% and 0.0899% respectively, the average errors are 0.000272% and 0.000117% respectively, and the STD of errors are 0.0063% and 0.0015% respectively. When more measurements are used, reconstruction errors can be reduced. It was found that if the number of measurements in each segment is less than 35, the reconstruction error can become large quickly. Therefore, there is a lower limit on the number of stored and transmitted measurements.

Three different bearing datasets collected in [121] are used to demonstrate the robustness of

the proposed framework. Training and testing signals from both of horizontal and vertical axes in the beginning, middle and near end of the life period are used, and the average reconstruction errors are compared in Table 9. It is found that the reconstruction errors for all periods and both axes are very small.

The proposed framework is compared with the traditional dictionary learning with K-SVD algorithm by randomly selecting time stamps for stored and transmitted measurements. With a total of 35 measurements in each segment, the reconstruction errors of the combined signal with the traditional K-SVD algorithm can be found in Figure 54. The maximum reconstruction error is 6.09%, the average error is 0.0196%, and the STD of errors is 0.1737%. It is seen that the proposed physics-constrained dictionary learning can significantly improve the reconstruction performance by optimizing the measurement matrix and the basis matrix separately.

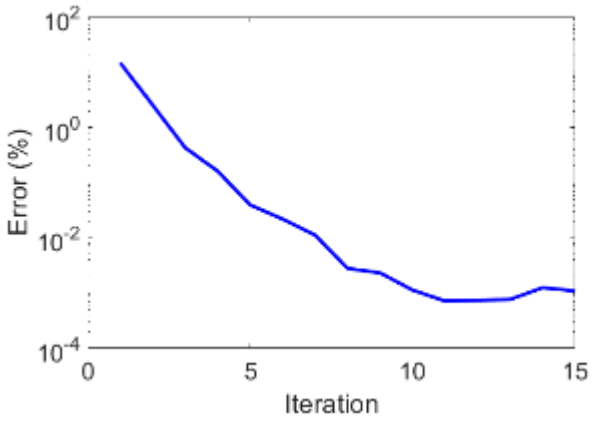


Figure 51. Convergence history of the reconstruction

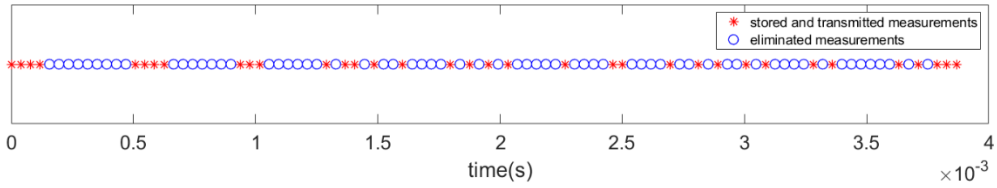
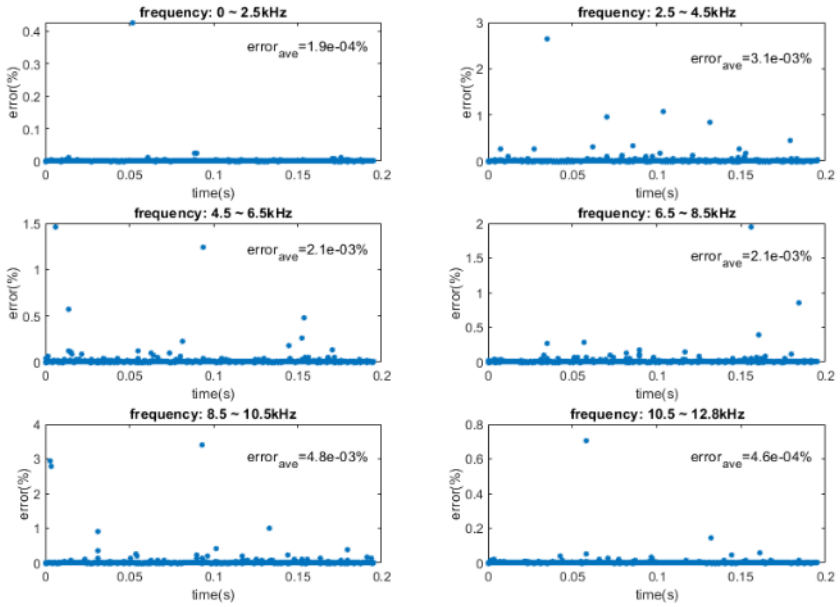
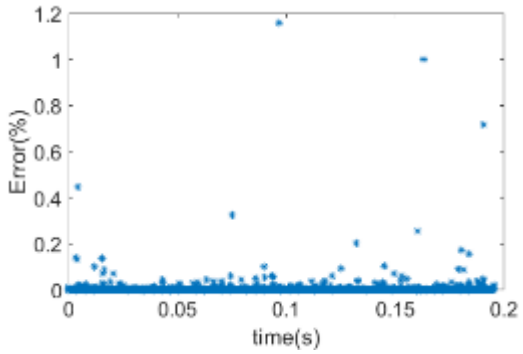


Figure 52. Optimized time stamps for data storage and transmission



(a) Reconstruction errors of decomposed signals



(b) Reconstruction errors of the combined signal

Figure 53. Reconstruction error of 50 testing segments

Table 9. Comparison of average reconstruction errors of signals collected for two axes during three different life periods in three datasets

Bearing dataset #	axis	Beginning	Middle	Near end
1	Horizontal	0.0033%	0.0034%	0.0024%
	Vertical	0.0034%	0.0026%	0.0025%
2	Horizontal	0.0039%	0.0018%	0.0036%
	Vertical	0.0038%	0.001%	0.00075%
3	Horizontal	0.0041%	0.0028%	0.003%
	Vertical	0.0034%	0.0022%	0.00073%

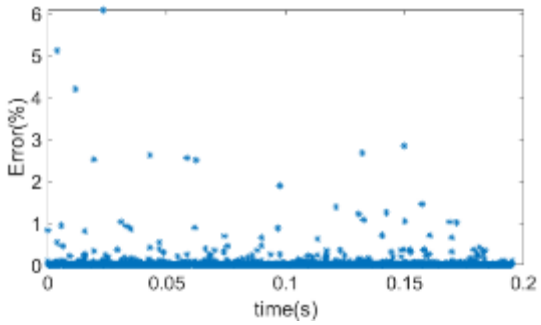


Figure 54. Reconstruction errors of the combine signal from 50 segments with traditional K-SVD

The target sparsity level s_l in Eq.(6.3) can affect the training process and reconstruction performance. With a smaller s_l, the original signal is reconstructed with fewer measurements. However, the training error of $\|\mathbf{S} - \mathbf{\Psi\Upsilon}\|_F^2$ can be large, because too few non-zero values in the coefficient vector can cause significant information loss and the original signal can no longer be represented with the basis matrix. Instead of assuming a constant s_l, the modified K-SVD can determine the desired s_l adaptively. Here, the result of the adaptive K-SVD is verified with the physics-constrained dictionary learning based on the conventional K-SVD when different values of s_l are assigned. The reconstruction errors from the conventional K-SVD with different s_l values

are listed in Table 10. For all tests, the number of measurements is 35. The size of the basis matrix $\boldsymbol{\Psi}$ is 100×300. It is seen in Table 10 that the reconstruction error has the lowest level when $s_l = 5$. When $s_l > 11$, the reconstruction error becomes very large, because the lower limit of the number of stored and transmitted measurements is not satisfied. More measurements are needed to reconstruct the original signal if $s_l > 11$. The convergence to the optimal s_l with the adaptive K-SVD algorithm is shown in Figure 55, which matches the trend in Table 10. The initial value of s_l is 1 in the adaptive K-SVD algorithm and adjustment step size Δs_l is 1. The incremental process is shown with the solid line. In the second test of the adaptive K-SVD, the initial value of s_l is 10. The decrease of sparsity level is shown as the dashed line. The average reconstruction error decreases as s_l decreases and the optimal $s_l$is also 5. Comparing results from conventional and modified K-SVD algorithms, the same desired s_l can be found which is $s_l = 5$. Since the computational cost is more expensive when s_l is larger because more non-zeros values in coefficient vectors need to be determined and higher sparsity level is preferred during the reconstruction process, initial s_l of 1 is suggested.

Table 10. Reconstruction errors with different sparsity levels

s_l	Max error (%)	Average error (%)	STD (%)
1	15.55	0.0072	0.2322
3	6.721	0.0029	0.0961
5	1.159	0.0024	0.0267
7	2.591	0.0035	0.0547
9	8.795	0.0369	0.2835
11	270.5	0.5521	6.0274

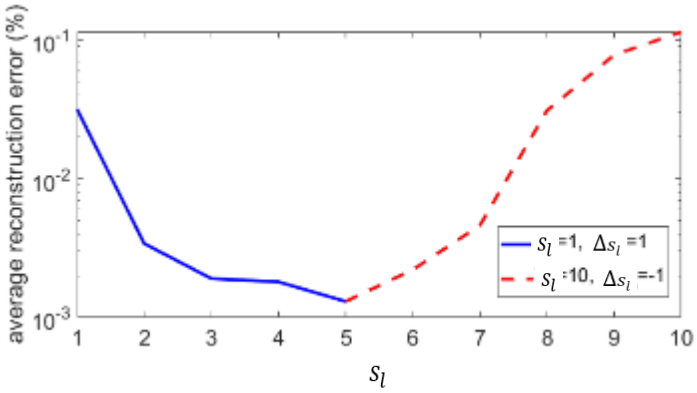


Figure 55. Convergence of s_l

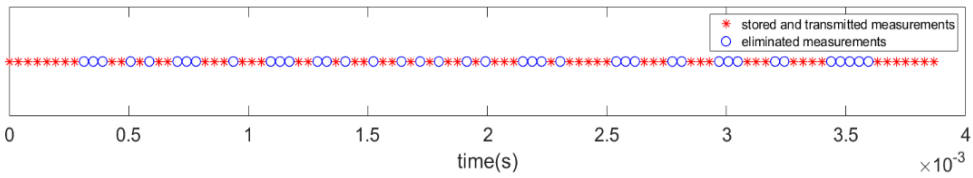
6.3.2 Experimental results with the physical constraint considered

In the second scenario, the constraint in Eq.(6.4) with the minimum sampling interval is considered. The sampling rate of the training signal is 25.6 kHz, and the sampling interval between consecutive data points is $r_s = 1/25600$ s. In this scenario, different minimum sampling interval of stored and transmitted data points are tested to optimize the measurement matrix. The minimum sampling interval is set to be r_s and $2r_s$ respectively.

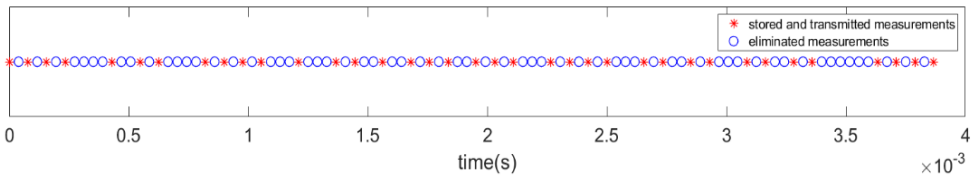
The initial number of stored and transmitted measurements in one segment is set to be 60 and optimized time stamps of measurements are indicated in Figure 56 (a) when the minimum sampling interval is r_s. It can be found that some stored and transmitted measurements are too close to each other. The collected information can be redundant. When the minimum sampling interval of $2r_s$ is used, some close-by measurements are eliminated as in Figure 56 (b). The 23 eliminated measurements are marked with circles and the 37 remaining data points to be stored and transmitted are marked as stars in Figure 56 (b).

The reconstruction errors with different minimum sampling intervals are shown in Figure 57. For the minimum sampling intervals of r_s and $2r_s$, the maximum reconstruction errors are

0.1499% and 0.9944%, the average errors are 0.0000773% and 0.0022%, and the STD of errors are 0.0022% and 0.0222% respectively. It is seen that the reconstruction errors increase when the sampling interval is increased, because fewer measurements are used. Compared to the reconstruction result in the first scenario in Section 6.3.1, where 35 measurements in each segment is used, the compression ratios between the two scenarios are similar. However, less redundant information is stored and transmitted with a higher level of reconstruction accuracy when the physical constraint is applied. The physical constraint of minimum sampling intervals is particularly useful when low cost sensors have limitations in sampling rates or there are bandwidth limitations in transmitting data in a distributed environment.

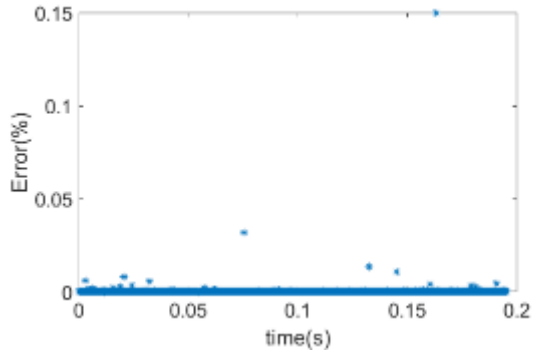


(a) Sampling interval is r_s

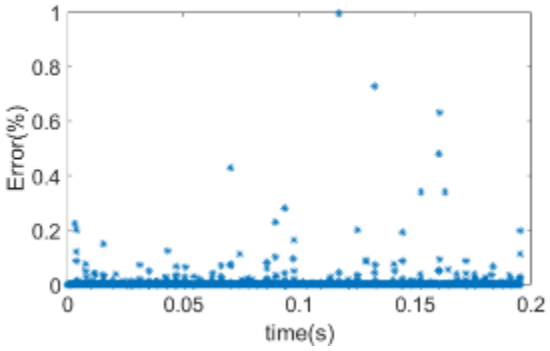


(b) Sampling interval is $2r_s$

Figure 56. Optimized measurements with different minimum sampling intervals as the physical constraint



(a) Sampling interval is r_s



(b) Sampling interval is $2r_s$

Figure 57. Reconstruction errors with different minimum sampling intervals as the physical constraint

6.4 Conclusion

A physics-constrained dictionary learning approach is presented to reduce the amount of data in storage and communication for 1D signals. Instead of all collected signals, only a few data samples are stored and transmitted, which can be used to reconstruct the complete signals. The energy consumption and memory usage in both data storage and communication can be saved. Different from other dictionary learning methods, the measurement matrix in the proposed physics-constrained dictionary learning formulation directly indicates the time stamps of samples or measurements. The basis matrix is optimized to sparsely represent the available data points. Compared to the conventional dictionary learning, the measurement and basis matrices are trained

separately in the physics-constrained dictionary learning, which significantly reduces the reconstruction errors. The minimum sampling interval is applied as an additional physical constraint to minimize the redundant information of measurements. The constrained FrameSense algorithm allows us to impose physical constraints during the training of measurement matrix. An adaptive K-SVD algorithm is also developed to train the basis matrix and determine the desired number of non-zero values in coefficient vectors on the fly. The proposed approach can be used to customize the basis matrices that target at decomposed signals with different bandwidths. In this way, the information loss due to compression can be minimized.

The major challenge of the proposed physics-constrained dictionary learning is related to the optimization in the high-dimensional space formed by the measurement and basis matrices. One limitation of the current computational scheme comes from the training algorithm that is based on the K-SVD. The K-SVD can only find the local optima. The performance of reconstruction depends on the choices of the initial basis matrix and the reconstruction algorithm. Instead of using the K-SVD, other dictionary learning algorithms can be applied. For instance, the online dictionary learning is more efficient than the K-SVD algorithm when monitoring real-time systems, because it does not need to store and access the entire dataset. The performance of reconstruction also depends on the dimensions of the dataset. High-dimensional data usually exhibit more correlations along multiple dimensions. Basis matrices may not be able to capture all, which affects the accuracy of reconstruction.

Here, the minimum sampling interval between compressed data points is used as the physical constraint. In order to store and transmit signals more efficiently, other application-specific physical constraints need to be considered. In large-scale sensor networks, the physical constraints can be designed based on the limitations of communication between sensors and the coverage of

the sensor network.

CHAPTER 7. Physics-Constrained Dictionary Learning for Manufacturing Process Monitoring with Imaging Systems

7.1 Introduction

A physics-constrained dictionary learning scheme based on Bayesian optimization (BO) is proposed to reduce the amount of data collection in manufacturing process monitoring with imaging systems. Instead of collecting the complete images, only a few pixel values at designed locations are collected and transmitted. With the limited data, the complete image can be reconstructed with CS. The measurement matrix and basis matrix are optimized separately and simultaneously for better performance. Additional physical constraints such as the coverage of measurements, the number of sensors and sensor accessibility need to be considered in the learning process. Here, additional physical constraints are applied for specific sensing strategies. When single-probe measurements (e.g. thermocouple and non-contact pyrometer) are used to measure temperatures at irregular spatial locations, the minimum distance between collected pixels is used as the additional physical constraint to reduce the redundancy of data acquisition in spatially correlated measurements. When a low-resolution camera is used to take measurements, the physical constraint that the distribution of the optimal locations of collected pixels needs to be close to the grid is considered. The new physics-constrained dictionary learning with different physical constraints is demonstrated with collected optical and thermal images in AM process monitoring.

The proposed physics-constrained dictionary learning algorithm based on BO is described in Section 7.2. The demonstration for the irregularly distributed measurements is given in Section 7.3. The demonstration for the measurement locations distributed as the grid is given in Section

7.4.

7.2 Methodology

The proposed physics-constrained dictionary learning optimizes the measurement and basis matrices simultaneously. For the optimized measurement matrix, there is only one non-zero entry in each row to determine the measurement locations. Additional physical constraints related to measurement locations and the number of measurements can be used to improve the efficiency of data collection. When the single-probe measurements are taken at the irregularly distributed pixel locations, the minimum distance between collected pixels is used as the physical constraint to minimize redundant information collected. When the low-resolution camera is used to take measurements, the physical constraint that the distribution of measurement locations needs to be close to the grid is applied. With a few collected pixels or measurements, the complete image can be reconstructed with CS. In Section 7.2.1, the framework of the proposed physics-constrained dictionary learning method is introduced. An improved physics-constrained dictionary learning method to further reduce the reconstruction errors of CS is illustrated in Section 7.2.2.

7.2.1 Formulation

The proposed physics-constrained dictionary learning is to optimize the measurement matrix $\mathbf{\Phi}$ and the basis matrix $\mathbf{\Psi}$ simultaneously under the physical constraints related to measurement locations. It is to solve the similar optimization problem in Eqs.(6.1-6.3), but the constraint in Eq.(6.4) is replaced with other physical constraints. In the first case, the optimal locations of collected pixels are irregularly distributed. Temperature readings at the optimal pixel locations can be obtained from the single-probe measurement. In the second case, the distribution of the optimal pixel locations needs to be close to the grid when the low-resolution camera is used to take

measurements.

Case 1: Single-probe measurements

In this case, single-probe measurements are used to measure pixel values or temperature readings at the optimal locations which are irregularly distributed. The minimum distance between collected pixels is used as the physical constraint. The physical constraint in Eq.(6.4) is replaced by

$$D_{ij}(\boldsymbol{\Phi}) \geq r, \quad \forall i, j \tag{7.1}$$

The constraint in Eq.(7.1) shows the physical limitation of collected pixels, such as the pair-wise distances between collected pixels need to be larger than a threshold value r.

The proposed learning procedure starts with an initial guess of the basis matrix $\boldsymbol{\Psi}$, which can come from some transformation matrices such as Fourier transformation, discrete cosine transformation, and wavelet transformation. In each iteration, two stages are performed to optimize the measurement and basis matrices. In the first stage, with the basis matrix $\boldsymbol{\Psi}$ fixed, the measurement matrix $\boldsymbol{\Phi}$ can be optimized by determining locations of collected pixels in each image. The optimization of pixel locations is solve based on the discrete Bayesian optimization (dBO) [122,123]. In the second stage, with the measurement matrix $\boldsymbol{\Phi}$ fixed, basis matrix $\boldsymbol{\Psi}$ can be optimized based on the K-SVD algorithm [84]. The above two optimization steps are repeated until both the optimal measurement and basis matrices converge without further improvement. Physical constraints such as the total number of pixels and the minimum distance between pixels collected can be incorporated.

7.2.1.1 Optimize the measurement matrix $\boldsymbol{\Phi}$ with the fixed basis matrix $\boldsymbol{\Psi}$

In the first stage, the basis matrix $\boldsymbol{\Psi}$ from the previous iteration is fixed, and the measurement

matrix $\boldsymbol{\Phi}$ is optimized based on the fixed $\boldsymbol{\Psi}$. Optimizing the measurement matrix $\boldsymbol{\Phi}$ is equivalent to searching for optimal locations of pixels for measurement or sampling. Here, the optimal locations are selected based on the dBO method.

Bayesian optimization (BO) is a robust global optimization scheme that incorporates uncertainty in the searching process. Different from other global optimization approaches such as the commonly used genetic algorithms, simulated annealing, and other heuristic algorithms, BO performs search based on a surrogate model, such as a Gaussian process regression (GPR) model, of the objective function. In addition, an acquisition function is constructed and used to guide the searching or sequential sampling process. It is designed to strike a good balance between exploration and exploitation. The searching process in BO can be accelerated with the properly designed surrogate model and acquisition function.

The dBO method here is an extension of BO to solve the combinatorial problems in an efficient way. The kernel function for the GPR model in dBO is defined as

$$k(\mathbf{z}, \mathbf{z}') = \exp(\textstyle\sum_{i=1}^{N} d(z_i, z_i')/\theta_i) \tag{7.2}$$

where θ_i's are the hyper-parameters of scales. The dBO method is used to solve

$$\min_{\mathcal{L}} \| \mathbf{S} - \boldsymbol{\Psi}\boldsymbol{\Upsilon} \| \tag{7.3}$$

$$\text{subject to } \boldsymbol{\Upsilon} = g(\boldsymbol{\Psi}, \boldsymbol{\Phi}(\mathcal{L}), \mathbf{S}) \tag{7.4}$$

$$|\mathcal{L}| = M \tag{7.5}$$

$$d\big(\mathbf{z}(\mathcal{L}_0), \mathbf{z}(\mathcal{L})\big) \leq p \tag{7.6}$$

where $\mathcal{L}$ indicate optimal locations and M is the target number of collected pixels The locations of collected pixels can be identified in the $M \times N$ measurement matrix $\boldsymbol{\Phi}(\mathcal{L})$ in a form of Eq.(3.12) where the column index of the value of 1 in each row indicates the location of each collected pixel in $\mathcal{L}$. The constraint in Eq.(7.4) is to recover $\boldsymbol{\Upsilon}$ from the linear equation $\boldsymbol{\Phi}(\mathcal{L})\mathbf{S} = \boldsymbol{\Phi}(\mathcal{L})\boldsymbol{\Psi}\boldsymbol{\Upsilon}$ with

the OMP algorithm. For the constraint in Eq.(7.5), the cardinality $|\mathcal{L}|$ meets the target number of data collection M. In Eq.(7.6), $\boldsymbol{z} = [z_1, \dots, z_N]$ where $z_i \in \{0,1\}$ $(\forall i = 1, \dots, N)$, is the binary string that indicates the pixel locations for data collection out of all possible N locations. The constraint is to limit the maximum deviation of the new solution $\boldsymbol{z}(\mathcal{L})$ away from the initial guess of optimal solution $\boldsymbol{z}(\mathcal{L}_0)$ measured by the hamming distance $d(\cdot)$. The acquisition function applied here is the expected improvement

$$a_{EI}(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^{D}, \theta) = \sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^{D}, \theta)\big(\gamma(\boldsymbol{x})\Phi\big(\gamma(\boldsymbol{x})\big) + \phi(\gamma(\boldsymbol{x}))\big) \tag{7.7}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density function and cumulative distribution function of the standard normal distribution, $\gamma(\boldsymbol{x}) = (\mu(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^{D}, \theta) - y_{best})/\sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^{D}, \theta)$ is the deviation away from the best solution y_{best} found so far, with posterior mean $\mu(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^{D}, \theta)$ and posterior standard deviation $\sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^{D}, \theta)$, given the existing D samples $\{\boldsymbol{x}_i, y_i\}_{i=1}^{D}$ and GPR hyper-parameter θ.

The performance of BO is dependent on the acquisition function and GPR surrogate. A good initial guess of the optimal solution provides a good starting point for searching with further improvement. The constraint in Eq.(7.6) helps reduce the search space for the combinatorial optimization problem. A good initial guess $\mathcal{L}_0$ can be obtained efficiently from the FrameSense algorithm by solving Eqs.(6.5) and (6.6) [120]. The FrameSense algorithm is to indirectly minimize $\text{MSE}(\widehat{\boldsymbol{\Upsilon}}) = \left\|\widehat{\boldsymbol{\Upsilon}} - \boldsymbol{\Upsilon}\right\|_{\mathbf{2}}^{\mathbf{2}}$, where $\widehat{\boldsymbol{\Upsilon}} = \underset{\boldsymbol{\Upsilon}}{\text{argmax}}\|\boldsymbol{\Psi}_{\mathcal{L}}\boldsymbol{\Upsilon} - \mathbf{S}_{\mathcal{L}}\|$ or $\widehat{\boldsymbol{\Upsilon}} = (\boldsymbol{\Psi}_{\mathcal{L}}^{*}\boldsymbol{\Psi}_{\mathcal{L}})^{-\mathbf{1}}\mathbf{S}_{\mathcal{L}}$. However, the FrameSense algorithm cannot guarantee that the recovered $\widehat{\boldsymbol{\Upsilon}}$ is also optimized in CS because an additional constraint in CS which is the sparsity level of $\widehat{\boldsymbol{\Upsilon}}$ needs to be considered. Therefore, with the solution from the FrameSense algorithm as the initial guess $\mathcal{L}_0$, the locations of collected pixels can be further optimized with the dBO method by solving Eqs. (7.3-7.6).

7.2.1.2 Optimize the basis matrix $\boldsymbol{\Psi}$ with the fixed measurement matrix $\boldsymbol{\Phi}$

In the second stage, the basis matrix $\boldsymbol{\Psi}$ can be optimized with the measurement matrix $\boldsymbol{\Phi}$ fixed by solving Eq.(6.7). With the physical constraint in Eq.(7.1) indicating the minimum Euclidean distance D_{ij} between each pair of collected pixel locations i and j, additional steps are needed to eliminate unsuitable pixel locations. Given optimized locations of collected pixels from the fixed basis matrix as the tentative locations, if the distance between any pair of the tentative locations is less than the threshold value r, one of them in the pair is eliminated and its pixel value is estimated by interpolating values of remaining collected pixels. The estimated pixel values at the eliminated locations are used to replace their original values in $\boldsymbol{\Phi}\mathbf{S}$. With $\mathbf{X} = \begin{pmatrix} \alpha\mathbf{S} \\ \boldsymbol{\Phi}\mathbf{S} \end{pmatrix}$, $\mathbf{Z} = \begin{pmatrix} \alpha\mathbf{I} \\ \boldsymbol{\Phi} \end{pmatrix}\boldsymbol{\Psi}$ is optimized by solving Eqs.(6.8) and (6.9) with the K-SVD algorithm [84]. $\boldsymbol{\Psi}$ can be solved with Eq.(6.13).

The proposed physics-constrained dictionary learning algorithm is shown in Table 11.

Table 11. Physics-constrained dictionary learning

Input:	Training signals $\mathbf{S}$; initial basis matrix $\mathbf{\Psi}_0$. sparsity s_l. minimum distance between collected pixel locations r; total number of iterations C; m=0;
Output:	Estimated sparse coefficients or parameters $\mathbf{\Upsilon}$; measurement matrix $\mathbf{\Phi}$; basis matrix $\mathbf{\Psi}$. Vector R for collected pixel locations.

Procedure:	WHILE $m < C$ DO
	1. Compute $\mathbf{\Phi}$ and the tentative pixel location (x, y) based on the dBO method.
	2. FOR i = 1 to the number of tentative pixel locations
	FOR j = 1 to the number of tentative pixel locations
	If $D_{ij}(\mathbf{\Phi}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r$, the pixel intensity at (x_i, y_i) is obtained from the measurement and the index i is stored in the vector R. The pixel value at (x_j, y_j) will be estimated by interpolating values of remaining collected pixels and the index j is stored in vector G.
	END FOR
	END FOR
	3. Values at locations with indices stored in G are estimated by extrapolating measurements with indices stored in R. Estimated values are used to replace the original values in the k rows of matrix $\mathbf{\Phi S}$, where k is the values in the vector G.
	4. With $\mathbf{X} = \begin{pmatrix} \alpha\mathbf{S} \\ \mathbf{\Phi S} \end{pmatrix}$, $\mathbf{Z} = \begin{pmatrix} \alpha\mathbf{I} \\ \mathbf{\Phi} \end{pmatrix}\mathbf{\Psi}$ and $\mathbf{\Upsilon}$ are updated by solving Eqs.(6.8) and (6.9) with K-SVD method.
	5. Update $\mathbf{\Psi}$ with Eq.(6.13)
	6. $m = m + 1$
	END WHILE

Case 2: Measurements with low-resolution camera

In this case, pixel values collected with the low-resolution camera are used to reconstruct the high-resolution images. In the training process, the high-resolution images are used to optimize the basis and measurement matrices. The distribution of optimized locations of collected pixels needs to be close to the regular grid of pixel locations in the low-resolution images. Therefore, the grid-like distribution of optimized locations as the physical constraint is applied in a modified FrameSense algorithm shown in Table 12. Compared to the original FrameSense algorithm, the

modified one applies the additional physical constraint in the step 3. If the unsuitable location belongs to the pixel locations in the low-resolution images, this location is remained in the set of optimized locations. With the pixel locations obtained from the algorithm in Table 12 as the initial guess, the dBO method can be used to further optimize pixel locations by solving Eqs.(7.3)-(7.6). The basis matrix and measurement matrix are then optimized with the algorithm in Table 11 without step 2 and step 3.

Table 12. The constrained FrameSense algorithm for 2D optical images

1. Initialize optimized measurement locations $\mathcal{L}$, all available measurement locations $\mathcal{N}$, all pixel locations of low-resolution images $\mathcal{A}$ and desired number of measurements m_t
2. Determine the first two removed rows in $\mathbf{\Psi}$ by solving $\mathcal{T} = \text{argmax}_{i,j\in\mathcal{N}} \left| < \boldsymbol{\psi}_i, \boldsymbol{\psi}_j > \right|^2$ and update remaining measurement locations $\mathcal{L} = \mathcal{N} \backslash \mathcal{T}$ by excluding $\mathcal{T}$
3. WHILE the length of $\mathcal{L} < m_t$ DO

    Find the i^*-th row in $\mathbf{\Psi}$ to eliminate by solving $i^* = \text{argmax}_{i\in\mathcal{L}} \text{F}(\mathcal{T} \cup \{i\})$, where $\text{F}(\mathcal{T} \cup \{i\})$ is the function in Eq.(6.5)
    WHILE $i^* \in \mathcal{A}$ DO
        Find the j^*-th row in $\mathbf{\Psi}$ to eliminate by solving $j^* = \text{argmax}_{j\in\mathcal{L}} \text{F}(\mathcal{T} \cup \{i\} \cup \{j\})$.
        $\mathcal{T} = \mathcal{T} \cup \{i\}$ and $i^* = j^*$
    END WHILE
    Update unsuitable measurement locations as $\mathcal{T} = \mathcal{T} \cup \{i^*\}$
    Update available measurement locations, $\mathcal{L} = \mathcal{L} \backslash \{i^*\}$
  END WHILE
4. Generate measurement matrix $\mathbf{\Phi}$ with optimized measurement locations $\mathcal{L}$

7.2.2 Improved physics-constrained dictionary learning

An improved physics-constrained dictionary learning algorithm is proposed to further reduce the reconstruction errors by taking the measurement at the position of the maximum reconstruction

error iteratively. The training data can be divided into two sub-datasets $\mathbf{S_1} = [\boldsymbol{s}_1, \boldsymbol{s}_2 \ldots \boldsymbol{s}_{\tilde{P}}] \in \mathbb{R}^{N\times\tilde{P}}$ and $\mathbf{S_2} = [\boldsymbol{s}_{\tilde{P}+1}, \boldsymbol{s}_{\tilde{P}+2} \ldots \boldsymbol{s}_P] \in \mathbb{R}^{N\times(P-\tilde{P})}$. The first sub-dataset $\mathbf{S_1}$ is used to optimize the basis matrix $\boldsymbol{\Psi}$ and measurement matrix $\boldsymbol{\Phi}$ with the algorithm in Table 11. With optimized $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$, the second sub-dataset $\mathbf{S_2}$ can be reconstructed with a few samples as $\boldsymbol{\Phi}\mathbf{S_2}$. The Frobenius norm of reconstruction errors of the second training sub-dataset is $e_{2\_norm} = \left\|\tilde{\mathbf{S}}_\mathbf{2} - \mathbf{S_2}\right\|_F$, where $\tilde{\mathbf{S}}_\mathbf{2}$ is the reconstructed second training sub-dataset . An additional measurement can be taken based on the position of the maximum reconstruction error by solving $(i^*, j^*) = \text{argmax}_{i,j}\left|([\tilde{\mathbf{S}}_\mathbf{2}]_{ij} - [\mathbf{S_2}]_{ij})/[\mathbf{S_2}]_{ij}\right|$, for $1 \le i \le N, 1 \le j \le P - \tilde{P}$. The pixel at the position of the maximum error is collected directly to reduce the overall reconstruction errors. Therefore, an additional row indicating the new measurement location is added to the original measurement matrix $\boldsymbol{\Phi}$ as a new measurement matrix $\widetilde{\boldsymbol{\Phi}}$ after each iteration. The basis matrix $\boldsymbol{\Psi}$ is then updated based on the new measurement matrix $\widetilde{\boldsymbol{\Phi}}$ as a new basis matrix $\widetilde{\boldsymbol{\Psi}}$. The detailed algorithm is in Table 13.

Table 13. Improved physics-constrained dictionary learning

Input:	First training sub-dataset $\mathbf{S_1}$; second training sub-dataset $\mathbf{S_2}$; basis matrix $\boldsymbol{\Psi}$ and measurement matrix $\boldsymbol{\Phi}$ from Table 11; maximum error allowed e_t
Output:	Measurement matrix $\widetilde{\boldsymbol{\Phi}}$; basis matrix $\widetilde{\boldsymbol{\Psi}}$.
Procedure:	WHILE $e_{2\_norm} > e_t$ DO 1. Obtain reconstructed second training sub-dataset $\tilde{\mathbf{S}}_\mathbf{2}$ with the basis matrix $\boldsymbol{\Psi}$ and measurement matrix $\boldsymbol{\Phi}$ by OMP algorithm 2. Compute e_{2_m} 3. Obtain the index of the maximum error (i^*, j^*) 4. Generate a $1 \times N$ zero row and replace the i^*-th zero in the row to one. Add the row to $\boldsymbol{\Phi}$. 5. With $\mathbf{X} = \begin{pmatrix}\alpha\mathbf{S_1} \\ \boldsymbol{\Phi}\mathbf{S_1}\end{pmatrix}$, $\mathbf{Z} = \begin{pmatrix}\alpha\mathbf{I} \\ \boldsymbol{\Phi}\end{pmatrix}\boldsymbol{\Psi}$ and $\boldsymbol{\Upsilon}$ are updated by solving Eqs.(6.8) and (6.9) with K-SVD. 6. Update $\boldsymbol{\Psi}$ with Eq.(6.13) END WHILD Output $\widetilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi}$, $\widetilde{\boldsymbol{\Psi}} = \boldsymbol{\Psi}$

7.3 Demonstration with single-probe measurements

The proposed physics-constrained dictionary learning was applied to monitor the temperature field in the AM process. A Hyrel 3D printer was used in the experiment to print a stair-like geometry with the size of 45mm×45mm×45mm as shown in Figure 58. The material was acrylonitrile butadiene styrene (ABS). During the printing process, a Seek thermal camera was used to capture the gray-scale image of the temperature field on one side face. The complete printing process takes about an hour and 1400 frames are collected. One frame to illustrate the experimental process is shown in Figure 59(a). The raw image includes insignificant information such as the extruder and printing bed. After performing image processing such as edge detection and background subtraction methods, the printed part can be extracted from the raw image as shown in Figure 59(b). The processed image is then rescaled to 10×10 pixels and the value of each pixel is corresponding to a temperature reading at the same location. In the traditional data acquisition method, the complete thermal image needs to be collected and stored, which is energy consuming and requires high memory usage if too many frames are captured. With the proposed data acquisition technique, only a few pixel values or temperature readings at optimal locations need to be collected. The complete thermal image can be reconstructed with these collected pixel values or temperature readings. Therefore, the data acquisition process can be more efficient. The complete training dataset is shown in Figure 60. Since the AM process is a repeatable manufacturing process, the testing dataset is generated by monitoring the manufacturing process of the same part. The same geometry of the part and machine settings are used when collecting training and testing datasets, so the level of similarity between two datasets is high. If the training dataset cannot be obtained from experiments, simulation models of the manufacturing process can

also be used to generate training images.

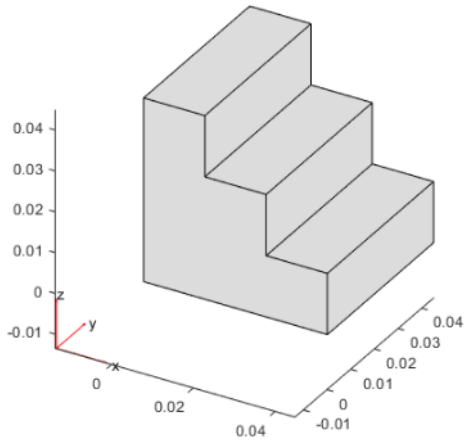
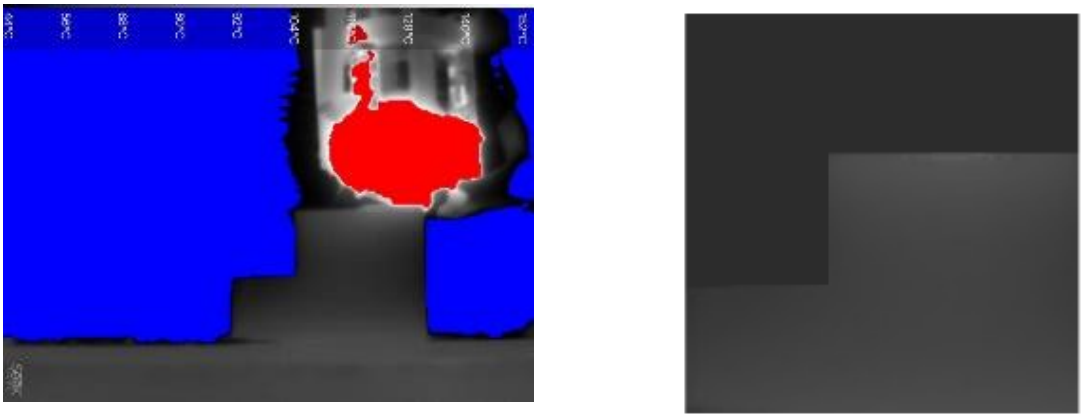


Figure 58. Stair-like geometry for printing



(a) Raw image

(b) Processed image

Figure 59. A captured frame during printing process before and after image processing

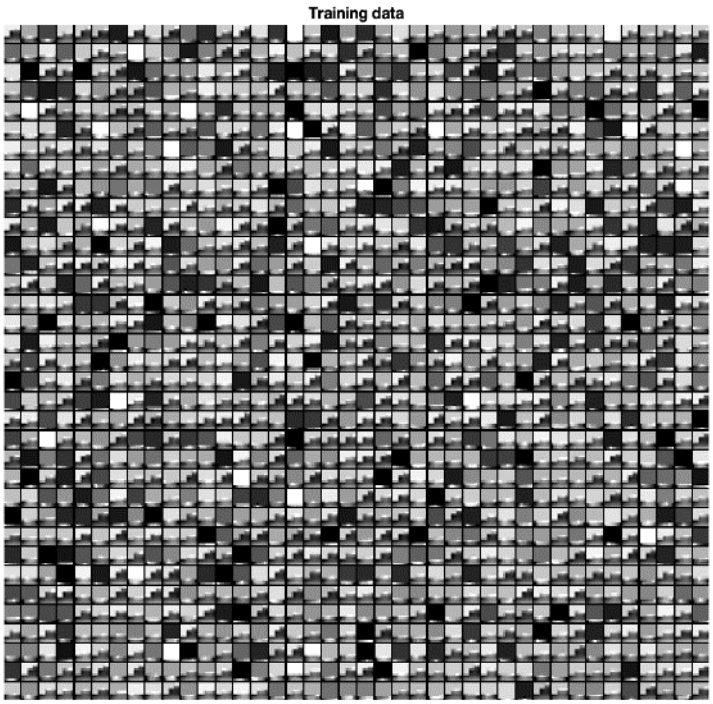


Figure 60. Training dataset including 1400 thermal images

Reconstruction performances with the physics-constrained dictionary learning method described in Table 11 and the improved physics-constrained dictionary learning method shown in Table 13 are compared in Section 7.3.1 and Section 7.3.2.

7.3.1 Reconstruction with physics-constrained dictionary learning

With the training dataset in Figure 60, the physics-constrained dictionary learning in Table 11 is used to design the measurement matrix $\mathbf{\Phi}$ and the basis matrix $\mathbf{\Psi}$. With optimized measurement and basis matrices, the OMP algorithm is performed to reconstruct the original images with a few pixel values. The physics-constrained dictionary learning approach is applied with and without considering the physical constraint in Eq.(7.1).

7.3.1.1 Physics-constrained dictionary learning without the physical constraint

In this scenario, the constraint in Eq.(7.1) that indicates the minimum distance between collected pixels is not considered. The size of the basis matrix $\boldsymbol{\Psi}$ is 100×300 and maximum number of non-zero values in each coefficient vector of $\boldsymbol{\Upsilon} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 \ldots \boldsymbol{\gamma}_P]$, s_l, is set to be 15. If a smaller s_l is used, the computational cost for the training process is reduced but the training errors are increased because sparse coefficient vectors with too few non-zero values are not representative for all training data with respect to the basis matrix. With 100 pixels or temperature readings in each thermal image, the number of collected pixel values is set to be 30. Locations of these 30 pixels are optimized with the physics-constrained dictionary learning in Table 11 without the constraint in Eq.(7.1). The initial basis matrix $\boldsymbol{\Psi}$ is the discrete cosine transformation matrix. The trained basis matrix is shown in Figure 61. The optimized locations of collected pixels are indicated with circles in Figure 62. It can be found that the distribution of optimal pixel locations is irregular and the IR thermal camera cannot be used to take measurements. Therefore, the single-probe measurement such as thermocouple and pyrometer needs to be used to measure temperatures at these pixel locations. A senor is used to take the measurement at each pixel location.

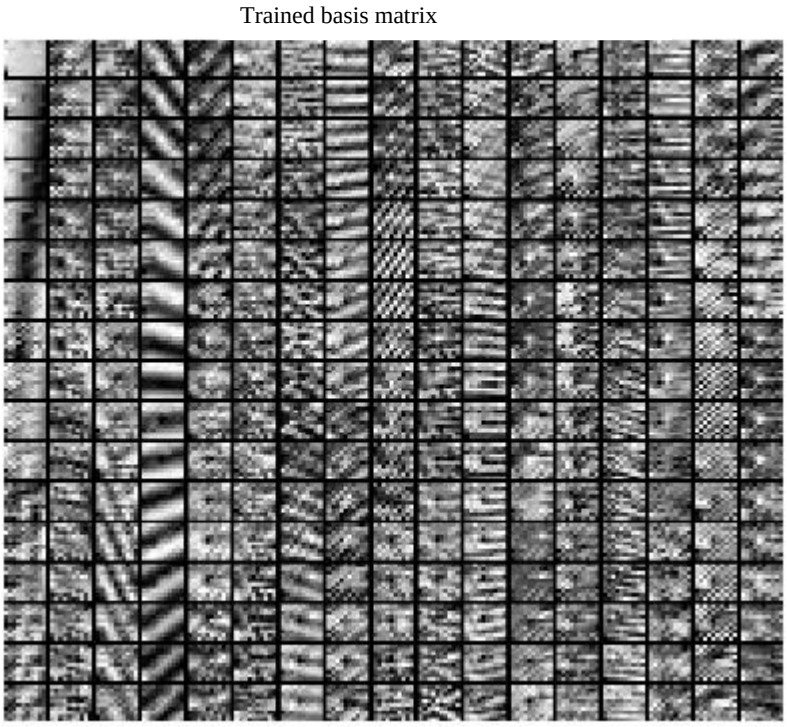


Figure 61. Optimized basis matrix

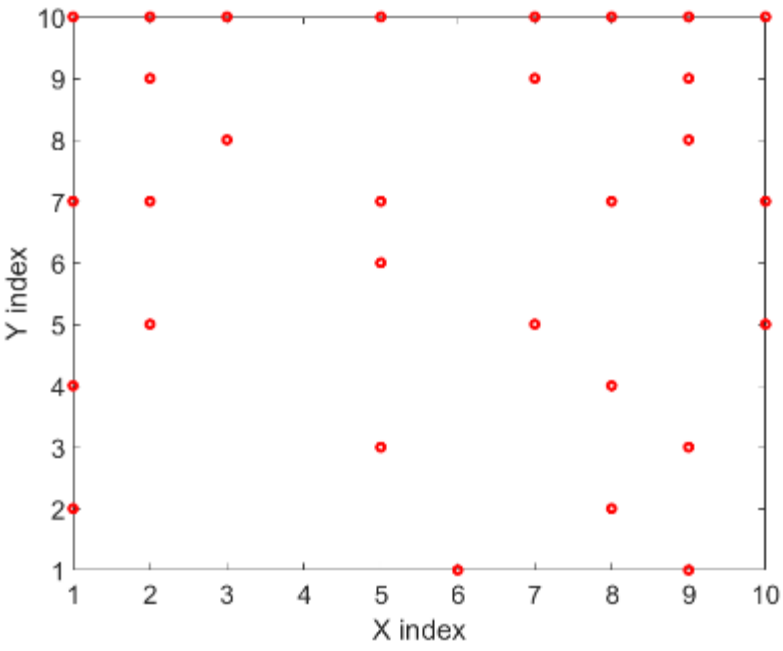
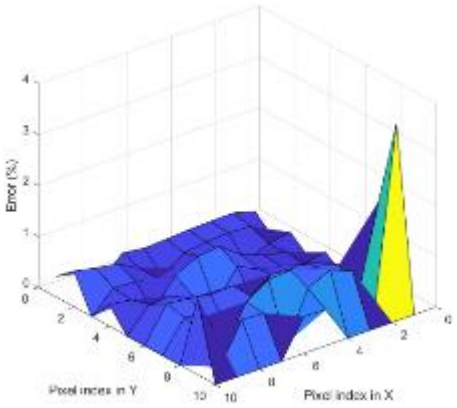
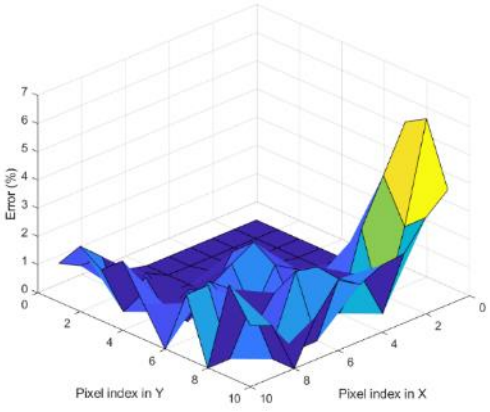


Figure 62. Optimized locations of collected pixels in the thermal image with 10×10 pixels.

With the optimized measurement matrix $\mathbf{\Phi}$ and basis matrix $\mathbf{\Psi}$, and 30 measurements in each thermal image, the testing data generated from 50 testing thermal images can be reconstructed with the OMP algorithm and the relative reconstruction errors are shown in Figure 36 (a). The error at each pixel location in Figure 36 (a) is the average error of 50 testing images at the same pixel location. It is found that the maximum error of 3.56 % is at the pixel location of (1, 9) in Figure 36 (a). The maximum error, the average error and the standard deviation of errors for overall reconstruction results from 50 testing images are 8.94 %, 0.42% and 0.71% respectively. The traditional K-SVD is also used to reconstruction 50 testing images with 30 measurements in each image. The size of the basis matrix $\mathbf{\Psi}$ and the sparsity level of the coefficient vector for K-SVD is set to be the same as those for the physics-constrained dictionary learning. However, the locations of 30 measurements are randomly selected for K-SVD. The reconstruction errors with K-SVD are shown in Figure 36(b). The maximum error, the average error and the standard deviation of errors of overall reconstruction results are 28.58%, 1.01% and 1.91% respectively. Therefore, it is found that the physics-constrained dictionary learning can significantly improve the reconstruction performance by optimizing the measurement and basis matrices simultaneously.



(a) Reconstruction errors of 50 testing images with physics-constrained dictionary learning



(b) Reconstruction errors of 50 testing images with K-SVD

Figure 63. Reconstruction errors with physics-constrained dictionary learning without considering the physical constraint, and reconstruction errors from K-SVD

A sensitivity analysis with different numbers of collected pixels is performed. Based on the results in Table 14, it is found that the reconstruction errors are reduced as more pixels are used. However, when more pixel values are collected, the physics-constrained dictionary learning method is less efficient for data collection. Therefore, 30 measurements are used as the trade-off between the reconstruction accuracy and the efficiency of data collection.

Table 14. Sensitivity analysis with different numbers of collected pixels

# of measurements	Maximum reconstruction error (%)	Average reconstruction error (%)	STD of reconstruction errors (%)
15	15.09	1.07	1.46
20	8.90	0.82	1.04
25	11.84	0.55	0.84
30	8.94	0.42	0.71
35	7.29	0.38	0.65
40	6.35	0.33	0.56

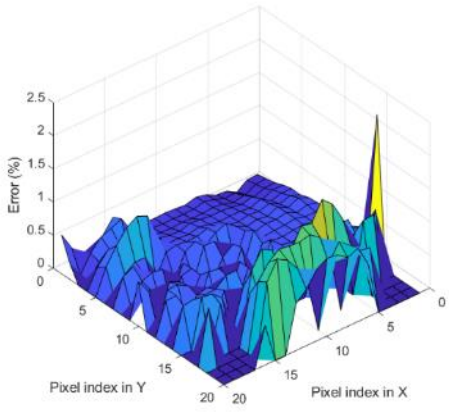
A sensitivity analysis with different numbers of training images is also performed. When 30 pixel values are collected in each image, the reconstruction errors with different numbers of

training images can be found in Table 15. When more training images are available and used, the reconstruction errors are reduced.

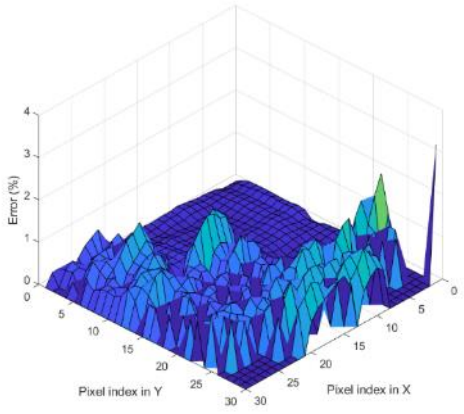
Table 15. Sensitivity analysis with different numbers of training images

# of training images	Maximum reconstruction error (%)	Average reconstruction error (%)	STD of reconstruction errors (%)
500	23.85	0.67	1.27
800	8.41	0.54	0.77
1100	9.42	0.46	0.75
1400	8.94	0.42	0.71

Here, the original image with a size of 238×219 is rescaled to 10×10 pixels to reduce the computational cost in the learning process. The proposed physics-constrained dictionary learning method can also be applied for images with larger sizes by keeping the same compression ratio. For rescaled images with 20×20 pixels, the same compression ratio of 3.33 is used, so that 120 pixels in each image are measured. The reconstruction errors are shown in Figure 64 (a), where the error at each pixel location is the average error of 50 testing images at the same location. The maximum error, the average error and the standard deviation of errors for overall reconstruction results from 50 testing images are 9.78 %, 0.34% and 0.58% respectively. For rescaled images with 30×30 pixels, 270 pixel values in each image are collected. The reconstruction errors are shown in Figure 64 (b). The maximum error, the average error and the standard deviation of errors for overall reconstruction results from 50 testing images are 10.59 %, 0.36% and 0.64% respectively. It is found that the proposed physics-constrained dictionary learning method is feasible for larger datasets, but the computational cost in the learning process can also be increased.



(a) Reconstruction errors for images with 20×20 pixels



(b) Reconstruction errors for images with 30×30 pixels

Figure 64. Reconstruction errors for different sizes of images

7.3.1.2 Physics-constrained dictionary learning with the physical constraint

To improve the efficiency of data collection, the constraint in Eq.(7.1) to indicate the minimum distance between collected pixel locations is considered. The minimum distance r between pixels is set to be 1.2. The optimized locations of collected pixels are indicated in Figure 65. Among 30 collected pixels in Section 7.3.1.1, as shown in Figure 62, six pixels marked with the triangular symbol in Figure 65 are not measured but estimated by interpolating the remaining pixels marked with the circular symbol. With the trained measurement matrix $\mathbf{\Phi}$ and basis matrix $\mathbf{\Psi}$, and 24 collected pixel values, the testing data can be reconstructed with OMP algorithm and the relative reconstruction errors are shown in Figure 66. The compression ratio is 4.17. The maximum error, the average error and the standard deviation of errors for overall reconstruction results from 50 testing images are 10.92 %, 0.49% and 0.91% respectively. Compared to the reconstruction results in Section 7.3.1.1 where 30 pixel values are collected and the physical constraint is not considered, the maximum and average of reconstruction errors increase about 2%

and 0.07% respectively because of the interpolation. However, the number of collected pixels is decreased from 30 to 24. Compared to the reconstruction results in Table 14 where 25 pixel values are collected, the maximum and average of reconstruction errors decrease about 1% and 0.06% respectively when the compression ratio is similar. Therefore, when the physical constraint is applied, the reconstruction accuracy and the efficiency of data collection can be improved by reducing the redundancy of collected information.

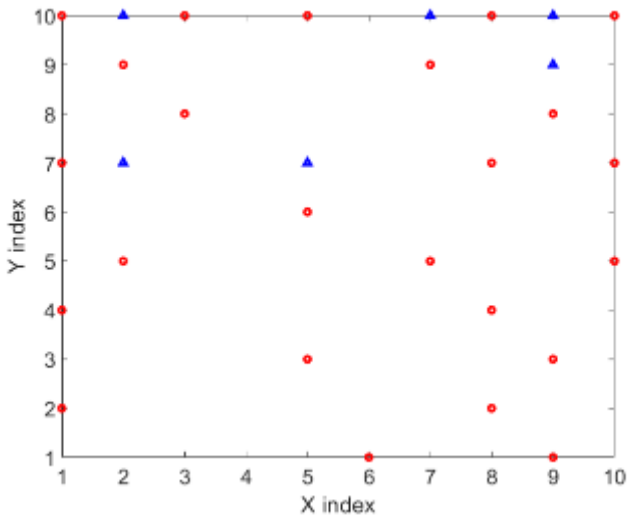


Figure 65. Optimized locations of collected pixels with physical constraint, where the values of pixels indicated by the triangular markers are not collected.

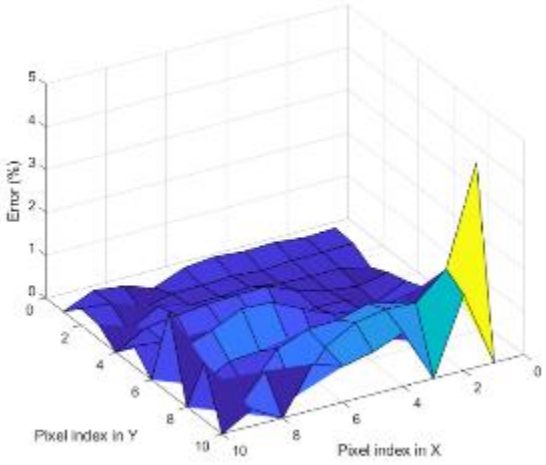


Figure 66. Reconstruction errors of 50 testing images with physics-constrained dictionary learning by considering the physical constraint.

7.3.2 Reconstruction with improved physics-constrained dictionary learning

To further reduce the reconstruction errors, the improved physics-constrained dictionary learning in Table 13 is performed. The original training dataset including 1400 images is separated to two sub-datasets. The first training sub-dataset has 1200 images and the second sub-dataset has 200 images. The maximum allowed norm of errors e_t is set to be 65. The first training sub-dataset is used for the algorithm in Table 11 to optimize the measurement matrix $\mathbf{\Phi}$ and the basis matrix $\mathbf{\Psi}$. With the optimized $\mathbf{\Phi}$ and $\mathbf{\Psi}$, the second sub-dataset is then used to determine the reconstruction errors with the OMP algorithm. The newly collected pixel location based on the index of the maximum reconstruction error is then added to $\mathbf{\Phi}$ iteratively until the Frobenius norm of reconstruction errors for the second sub-dataset is less than e_t.

In Section 7.3.2.1, improved physics-constrained dictionary learning is applied without considering the physical constraint. The physical constraint in Eq.(7.1) is incorporated into the improved physics-constrained dictionary learning method in Section 7.3.2.2.

7.3.2.1 Improved physics-constrained dictionary learning without the physical constraint

In the scenario when the constraint in Eq.(7.1) is not considered, the initial number of collected pixel values is set to be 25. With the improved physics-constrained dictionary learning, five additional pixel values based on the index of the maximum reconstruction error are collected. With a total of 30 pixel values collected, the reconstruction errors form the 50 testing images are shown in Figure 67. The compression ratio is 3.33. It is found that the maximum error of 1.49 % is at the pixel location of (3, 8) in Figure 67. The maximum error, the average error and the standard deviation of errors of overall reconstruction results from 50 testing images are 6.11 %, 0.40% and

0.62% respectively. Compared to results in Section 7.3.1.1 when the physics-constrained dictionary learning is applied with 30 measurements, the improved physics-constrained dictionary learning further reduces the reconstruction errors because the pixel at the location of the maximum reconstruction error is directly collected.

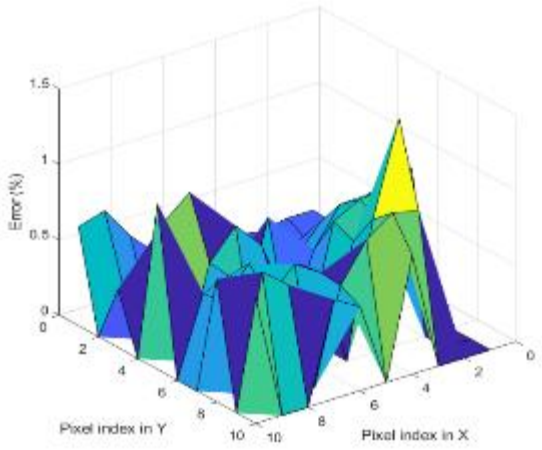


Figure 67. Reconstruction errors when the physical constraint is not considered with improved physics-constrained dictionary learning

7.3.2.2 Improved physics-constrained dictionary learning with the physical constraint

In the scenario when the constraint in Eq.(7.1) is considered and the minimum distance r is 1.2. The algorithm in Table 13 is used and the termination criteria here is when the number of tentatively collected pixel values reaches 30 in order to compare with the reconstruction performance in Section 7.3.2.1. Therefore, among 30 tentative pixel values, 24 pixel values are directly collected, and 6 pixel values are estimated by interpolating the values of 24 collected pixels. With a total of 24 collected pixels, the reconstruction errors of 50 testing images are shown in Figure 68. The compression ratio is 4.17. It is found that the maximum error of 1.65 % is at the

pixel location of (2, 8) in Figure 68. The maximum error, the average error and the standard deviation of errors for overall reconstruction results from 50 testing images are 8.39%, 0.44% and 0.74% respectively. After introducing the constraint in Eq.(7.1), the reconstruction errors are increased, but the number of collected pixel values required is reduced from 30 to 24. Compared to results in Section 7.3.1.2 where the physics-constrained dictionary learning is applied with the physical constraint, the reconstruction errors are reduced with the improved physics-constrained dictionary learning. The reconstruction performances in different scenarios are shown in Table 16.

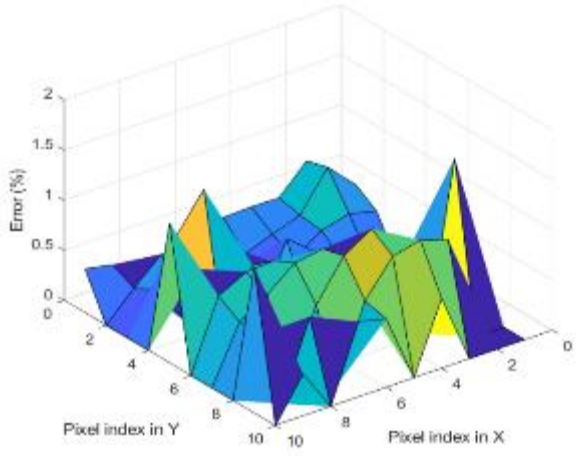


Figure 68. Reconstruction errors when the constraint in Eq.(7.1) is considered with improved physics-constrained dictionary learning

Table 16. Reconstruction performance with different scenarios

	physics-constrained dictionary learning		improved physics-constrained dictionary learning	
With/without the physical constraint in Eq.(7.1)	Without	With	Without	With
Number of measurements	30	24	30	24
Compression ratio	3.33	4.17	3.33	4.17
Maximum reconstruction error	8.94%	10.92 %	6.11 %	8.39%
Average reconstruction error	0.42%	0.49%	0.40%	0.44%
Standard deviation of reconstruction errors	0.71%	0.91%	0.62%	0.74%

7.4 Demonstration with measurements from low-resolution camera

In Section 7.3, the optimal locations of collected pixels are irregular. The single-probe measurement is used to measure temperatures at pixel locations. However, if the number of optimal locations become larger, this can be inefficient. If the low-resolution IR thermal camera is available, temperature readings at various pixel locations can be measured simultaneously. When pixel values collected by the low-resolution IR thermal camera are used to reconstruct the high-resolution images, the distribution of the optimal pixel locations in the high-resolution images needs to be close to the regular grid pixel locations in the low-resolution images. Temperature readings at locations between the grid points cannot be directly measured. Therefore, the constrained FrameSense algorithm in Table 12 is used to optimize the locations of collected pixels, and the physical constraint that the distribution of optimal locations needs to be close to the grid

is considered. In Section 7.4.1, the laser beam spot and powder spatter in the SLM process is monitored with a high-speed optical camera. The low-resolution pixel values captured by the optical camera are used to reconstruct the high-resolution images. The regions of interest such as the laser beam spot and powder spatter are difficult to observe in the low-resolution images. Our goal is to identify these features in the reconstructed high-resolution images. In Section 7.4.2, the low-resolution pixel values or temperature readings collected with the IR thermal camera are used to reconstruct the high-resolution thermal images. The accuracy of each pixel value or temperature reading in the reconstructed high-resolution thermal images is of interest. Experimental results in Section 7.3 are used for demonstration.

7.4.1 Reconstruction of optical images

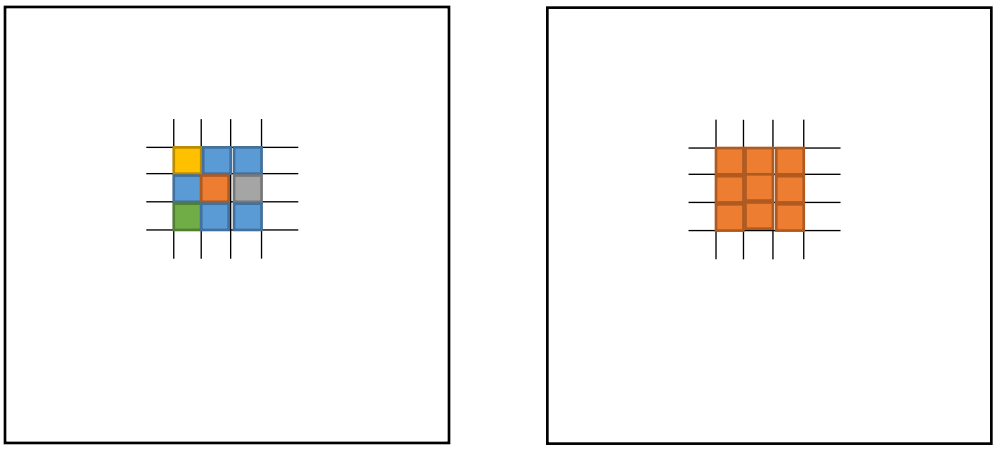
In the SLM process, the shape and size of the melt pool need to be controlled. The grain structures can vary and defects can be formed during the rapid solidification when the size of the melt pool is too large or too small. The size of the laser beam spot is one of the important factors affecting the geometry of the melt pool. Powder spatter during the manufacturing process is another major cause of defect formation. Spatter can also cause contamination in the powder bed. Therefore, monitoring the laser beam spot and powder spatter in the SLM process is critical to control the quality of the solid build.

A total of 1700 images are collected with a high-speed optical camera. One example is shown in Figure 69. From the high-resolution image with the size of 25×25 pixels in Figure 69(b), the shape of the laser beam spot enclosed by the oval with solid line and powder spatter in the region enclosed by the circle with dashed line can be visualized. However, the shape of the laser beam spot and powder spatter are hard to be identified from the low-resolution image with the size of 9×9 pixels in Figure 69(a). The relation between pixels of the high-resolution and low-resolution

images is shown in Figure 70. From the high-resolution image in Figure 70(a), the low-resolution image in Figure 70(b) is obtained by replacing values of neighbouring pixels with the value at the center pixel. Our goal here is to reconstruct high-resolution images with features of the laser beam spot and powder spatter based on the low-resolution images.

(a) Low-resolution image (b) High-resolution image

Figure 69. High-resolution and low-resolution image for the laser beam spot and powder spatter



(a) High-resolution image (b) Low-resolution image

Figure 70. Relation between high-resolution and low-resolution images.

To reconstruct the high-resolution images from the low-resolution resolution ones, two approaches are compared. The first approach is to obtain the high-resolution images by interpolating all available pixel values in the low-resolution images. The second approach is to

reconstruct the high-resolution images with CS and the proposed physics-constrained dictionary learning method.

For the physics-constrained dictionary learning method, 1650 images are used for the training process and the remaining 50 images are used for testing. The algorithm developed for the second case in Section 7.2.1 is used to optimize the measurement and basis matrices. The physical constraint that the distribution of optimal locations needs to be close to the grid is applied. The size of the basis matrix $\mathbf{\Psi}$ is 625×700 and maximum number of non-zero values in each coefficient vector is set to be 65. The total number of collected pixels is 94. Instead of measuring all pixel values at the optimal locations, only ones at grid points are measured with the low-resolution camera. Other pixel values at locations between grid points are estimated by linearly interpolating measured ones. With measured and estimated pixel values, the high-resolution image can be reconstructed with the proposed physics-constrained dictionary learning method. The pixel locations optimized with the physics-constrained dictionary learning method are shown in Figure 71. The values at locations marked with the square symbol are estimated by linearly interpolating all pixel values marked with the circle symbol, which are obtained directly from the low-resolution image. With the optimized measurement and basis matrices, the high-resolution images are reconstructed with the OMP algorithm. Among the 50 original high-resolution images for testing, 8 of them are shown in Figure 72 (a). The reconstructed high-resolution images with simple linear interpolation are shown in Figure 72 (b), and the reconstructed results based on the proposed physics-constrained dictionary learning and CS are shown in Figure 72 (c). Compared to the results from the linear interpolation, the reconstructed high-resolution images with the physics-constrained dictionary learning method show the laser beam spot and powder spatter more clearly.

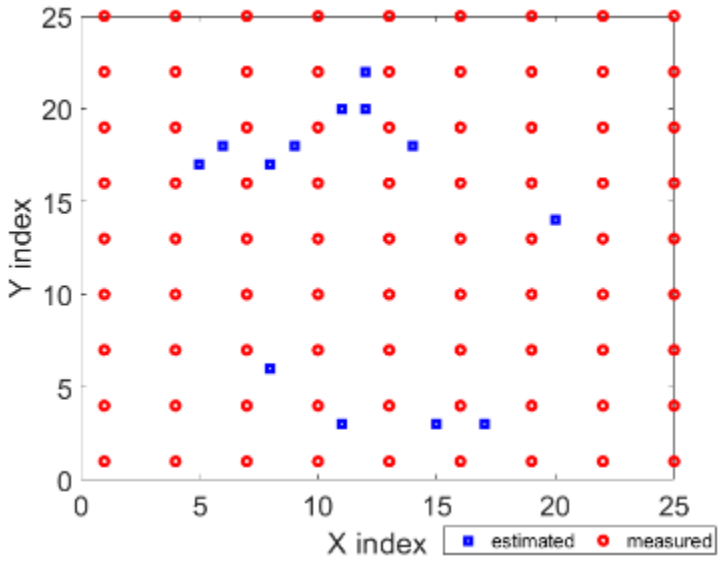
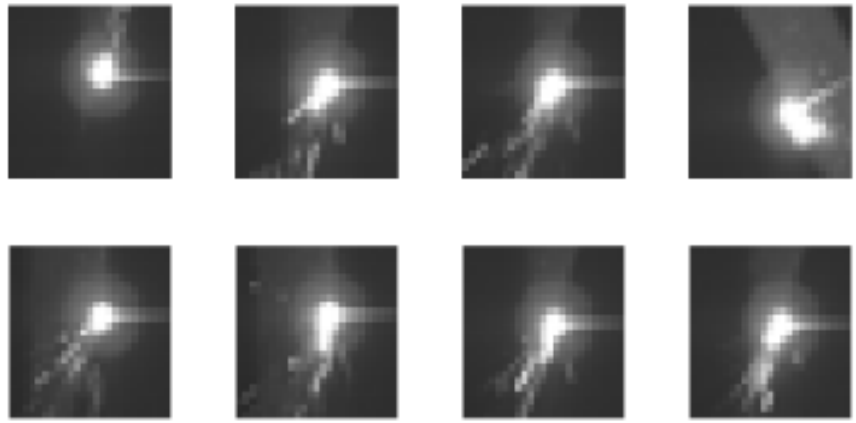
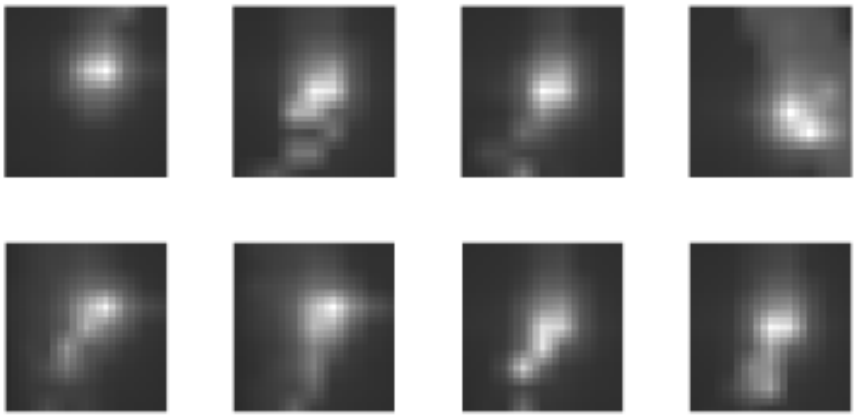


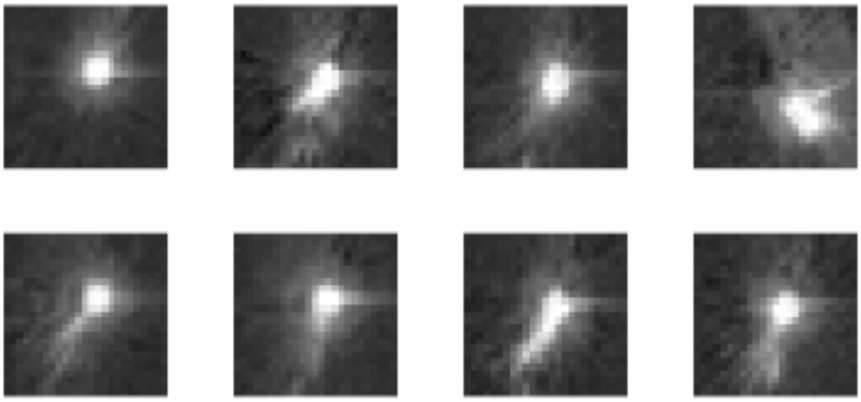
Figure 71. Optimized locations of collected pixels in the optical image with 25×25 pixels



(a) Original high-resolution images



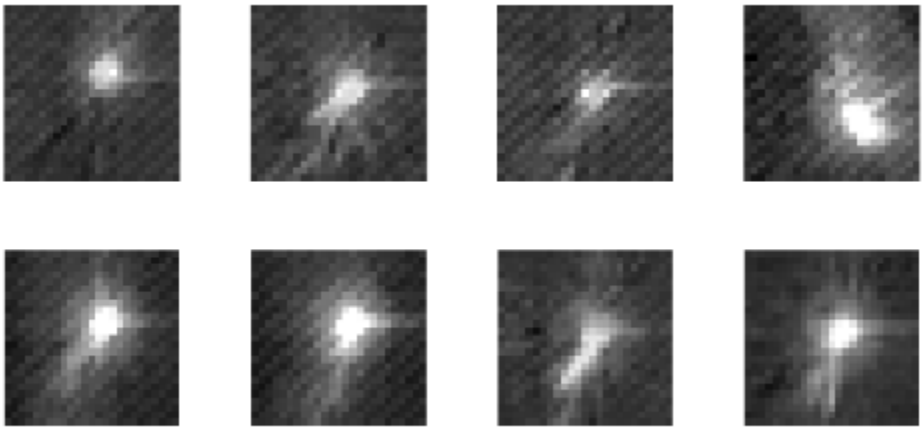
(b) Reconstructed high-resolution images by interpolation



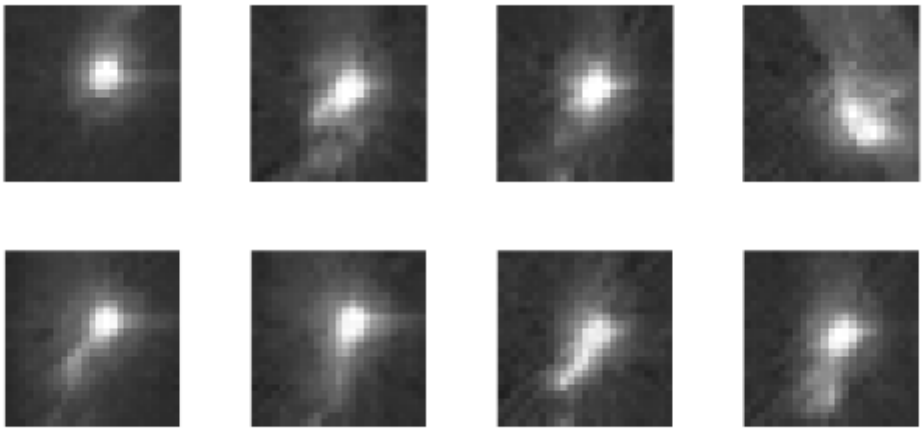
(c) Reconstructed high-resolution images by CS and physics-constrained dictionary learning

Figure 72. Comparison of reconstructed results between simple interpolation and the physics-constrained dictionary learning approach.

A sensitivity analysis is also performed to analyse the effect of the number of collected pixel values on the reconstruction performance. With CS and physics-constrained dictionary learning, the high-resolution images reconstructed with 81 and 133 collected pixel values are shown in Figure 73(a) and (b) respectively. Compared to the results in Figure 72(c) where 94 pixel are collected, reconstructed images in Figure 73(a) contain more noises. However, when more pixel values are collected, reconstructed images can also be more blurred as shown in Figure 73(b), because more interpolation errors are included in the reconstruction results when more estimated pixel values between grid points are used. Therefore, the trade-off between noises and interpolation errors needs to be considered when the number of collected pixel values is selected.



(a) Reconstructed high-resolution images with 81 collected pixels



(b) Reconstructed high-resolution images with 133 collected pixels

Figure 73. Reconstructed results with different numbers of collected pixels by CS and physics-constrained dictionary learning.

7.4.2 Reconstruction of thermal images

For optical images, identifying features such as the shape of the laser beam spot and powder spatter in images is important, whereas the accuracy of reconstructed pixel values are not critical. In contrast, for thermal images such as experimental results in Section 7.3.1, the accuracy of each pixel value or temperature reading in the reconstructed images needs to be considered. Therefore,

to minimize the overall reconstruction errors of thermal images, an additional physical constraint is required. Since values of collected pixels between grid points in the low-resolution image are estimated by linear interpolation, the interpolation error can be large near the high temperature gradient region. Based on the experimental results in Section 7.3.1, the high temperature gradient region is the boundary of the part, as indicated by the dash line in Figure 74. Therefore, the additional physical constraint is to avoid collecting pixel values at the boundary of the part to reduce the interpolation error. The boundary can be predicted based on the gradient of pixel values in the low-resolution thermal image. To consider this physical constraint, an additional step $\mathcal{L} = \mathcal{L} \backslash (\mathcal{L} \cap \mathcal{M})$ is added between step 3 and step 4 for the algorithm in Table 12, where $\mathcal{M}$ includes pixel locations at the boundary of the part in the low-resolution image. Since the boundary of the part is dynamically changing in the AM process, the training process needs to be repeated for each collected low-resolution image. In the application of AM process monitoring, the dynamic change of the boundary of the part can be obtained from the CAD model and the predefined tool path, and the training process can be done in advance for all collected low-resolution images.

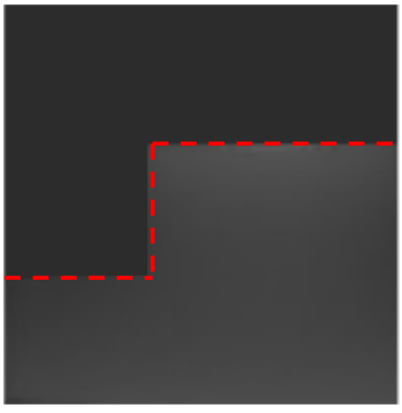


Figure 74. High temperature gradient at the boundary of the part.

The training dataset and testing dataset are obtained by rescaling the processed images in Section 7.3 to 13×13 pixels. The high-resolution thermal images with the size of 13×13 pixels can be reconstructed with the low-resolution ones with the size of 7×7 pixels. The physics-constrained dictionary learning method with two constraints is used to optimize the measurement and basis matrices. First, the distribution of the optimal locations need to be as close to the grid as possible. Second, the optimal locations of collected pixels should not be at the boundary of the part. The optimized measurement locations for one of the testing images are shown in Figure 75. There are 102 optimal locations. The 49 optimal locations marked as the circle symbol can be measured directly with the low-resolution camera, and the pixel values at these locations are used to estimate the values at the remaining 53 optimal locations, marked as the square symbol, by linear interpolation.

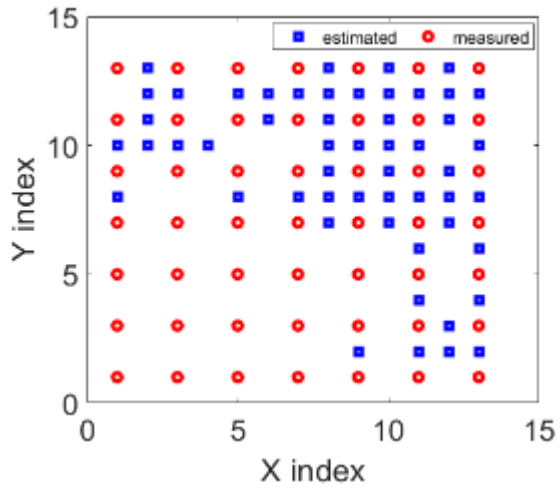
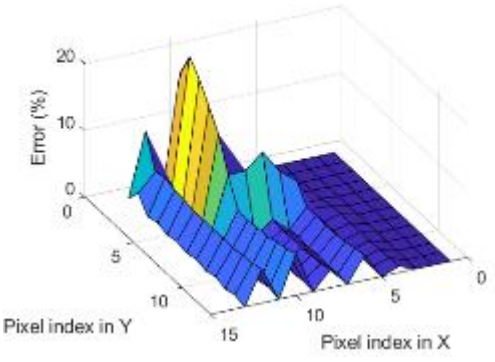


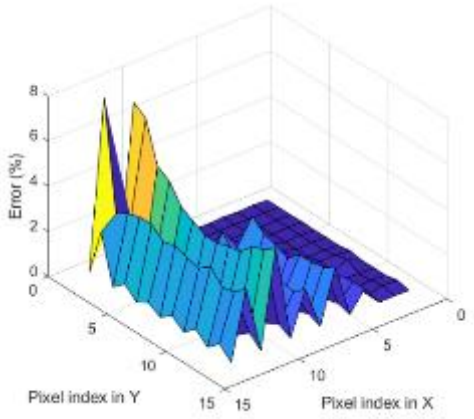
Figure 75. Optimized locations of collected pixels in the thermal image with 13×13 pixels.

The reconstruction results from the proposed approach and simple linear interpretation are compared. For simple linear interpretation, the high-resolution images are reconstructed by linearly interpolating all available pixel values in the low-resolution images. The relative

reconstruction errors are shown in Figure 76 (a). The error at each pixel location in Figure 76 (a) is the average error of 50 testing images at the same pixel location. It is found that the maximum error is 20.88 % in Figure 76 (a). The maximum error, the average error and the standard deviation of errors for overall reconstruction results from 50 testing images are 47.44 %, 1.79% and 5.31% respectively. In the proposed approach, the high-resolution images are reconstructed with CS based on the physics-constrained dictionary learning method. With the optimized measurement matrix $\mathbf{\Phi}$ and basis matrix $\mathbf{\Psi}$, the testing data generated from 50 testing thermal images can be reconstructed with the OMP algorithm and the relative reconstruction errors are shown in Figure 76 (b). The maximum error with respect to pixel locations is 7.45 % in Figure 76 (b). The maximum error, the average error and the standard deviation of errors for overall reconstruction results from 50 testing images are 19.78 %, 0.86% and 2.2% respectively. Compared to the results from interpolation, the reconstruction errors are significantly reduced with CS based on the physics-constrained dictionary learning method.



(a) Reconstruction errors of 50 testing images with linear interpolation



(b) Reconstruction errors of 50 testing images with physics-constrained dictionary learning

Figure 76. Reconstruction errors with simple liner interpolation and physics-constrained dictionary learning for thermal images

7.5 Conclusion

A physics-constrained dictionary learning approach is proposed to reduce the amount of pixels collected in manufacturing process monitoring with imaging systems. Instead of collecting the complete images, a few pixels are collected and can be used to reconstruct the complete images. The energy consumed in data collection and memory usage are then saved. The measurement and basis matrices are optimized in two stages. In the first stage, with the basis matrix $\mathbf{\Psi}$ fixed, the measurement matrix $\mathbf{\Phi}$ can be optimized by determining locations of collected pixels in each image. The locations of collected pixels are optimized based on the dBO method. In the second stage, with the measurement matrix $\mathbf{\Phi}$ fixed, basis matrix $\mathbf{\Psi}$ can be optimized based on the K-SVD algorithm. Different physical constraints are applied when different sensors are available. If the single-probe sensor can be used to take measurements, the distribution of optimal locations of collected pixels can be irregular and the minimum distance between collected pixel locations is performed as an additional physical constraint to minimize redundant information collected. The compression ratio is improved from 3.33 to 4.17. However, the maximum error is also increased about 1% and the average error is increased about 0.06% because of uncertainties from the interpolation. If the low-resolution camera is used in experiments, the physical constraint that the distribution of optimal locations needs to be close to the grid is applied. To improve the accuracy of each reconstructed pixel value, the additional physical constraint is imposed to eliminate pixel locations at the boundary of the part where the interpolation errors are large. The trade-off between the number of pixels collected and reconstruction errors needs to be considered to improve the data acquisition procedure. The reconstruction error can be further reduced by collecting the additional pixel at the location of the maximum reconstruction error from the training sub-dataset iteratively.

In monitoring the temperature distribution of the FFF process in CHAPTER 3 and the thermofluid field of the melt pool in the SLM process in CHAPTER 5, temperature measurements on the surface are required to reconstruct the complete temperature distribution or thermofluid field with the proposed PBCS approach. These measurements can be taken with single-probe sensors or low-resolution thermal cameras. However, as more measurements used, PBCS reconstruction can be more accurate. The physics-constrained dictionary learning approach proposed here can be used to recover the high-resolution images from single-probe measurements or low-resolution images without additional measurements. With the recovered high-resolution images, the PBCS can be further applied and the reconstruction accuracy can be improved because more temperature readings on the surface are available.

CHAPTER 8. Conclusions

In this dissertation, a novel physics-based compressive sensing approach is presented to monitor AM processes. Different from traditional CS techniques without the consideration of application domains, the proposed PBCS approach improves the compression ratio and reconstruction accuracy, which takes advantages of the domain knowledge in applications as physical models. The PBCS enables us to reconstruct 3D temperature and velocity fields in AM processes with higher temporal and spatial resolutions based on a limited amount of data collection. The complete 3D fields usually cannot be measured directly from conventional sensors. PBCS recovery and reconstruction of 3D fields can be done with only a few measurements on the surface. Therefore, PBCS provides a new way to obtain unobservable quantities of interest from the ones which can be easily measured. PBCS is an efficient approach to collect data from sensors by building connections between physical and digital spaces. From limited measurements, PBCS can be used to extract more information from data such as the load vectors including boundary conditions than pure statistical correlations. Compared to conventional sensors for process monitoring, PBCS is more efficient by reducing the amount of data collection without sacrificing the amount of information exchanged. The costs of data in communication and storage are reduced. Based on different physical models, the applications of PBCS to monitor different physical quantities have been demonstrated in this research.

8.1 Summary of the work

8.1.1 PBCS for temperature monitoring

In monitoring the temperature distribution of the FFF process, the PBCS formulation based on a heat transfer model is proposed. The reconstruction performances with different measurement

strategies such as the single-probe measurement and low-fidelity measurement are compared. It is found that the reconstruct accuracy is improved when more measurements are used. A new constrained OMP algorithm is also developed which can predict non-zero coefficient indices in the recovered sparse vector by using physical knowledge as the additional constraint. The constrained OMP algorithm is especially useful to improve the performance of PBCS. The computational cost in the recovery process is reduced with a domain decomposition method, which only uses a portion of nodes to construct matrices with smaller sizes in the finite element model. GPR is also used to predict and compensate PBCS systematic errors for more accurate reconstruction.

8.1.2 PBCS for flow velocity field monitoring

The PBCS formulations based on fluid flow models are also developed to monitor velocity fields in laminar and turbulent flows. The fluid flow models for laminar and turbulent flows are built based on the Navier-Stokes equations and the Reynolds Averaged Navier-Stokes equations respectively. For the laminar flow model, the load vector in the equation of the conservation of momentum is first recovered and the complete velocity field can be reconstructed with the forward modelling. For the turbulent flow model, with the recovered load vector in the equation of the conservation of momentum, the velocity field, the turbulent kinetic energy, and the dissipation rate are reconstructed simultaneously. The PBCS mechanism based on the turbulent flow model has been demonstrated with the steady-state backward-facing step flow. With a few velocity measurements on boundaries, the complete flow velocity field is reconstructed with PBCS. It is found that the regions of interest, especially those with high gradients, may require more measurements to maintain a reasonable accuracy level.

8.1.3 PBCS for temperature distribution and melt flow monitoring

The heat transfer model and the fluid flow model are then coupled as a multi-physics thermofluid model for PBCS to monitor the temperature distribution and melt flow of the melt pool in the SLM process. The reliability of the physical model is critical for the PBCS performance. Therefore, experimental results are used to calibrate parameters such as the model dimension, the mesh size and solver techniques for the thermofluid model. By using real-time measurements to recover the boundary conditions of transient models such as the load vector of the heat transfer model, the 3D temperature distributions and velocity fields can be reconstructed with high temporal and spatial resolutions, which overcomes the limitations of conventional sensors to measure temperature and velocity fields.

The sensing performance is further improved with a physics-constrained dictionary learning approach. The physics-constrained dictionary learning approach can optimize the basis and measurement matrices in CS simultaneously. Additional physical constraints such as the coverage of measurements, the number of sensors, sensor accessibility, and the energy consumption of data collection are considered in the learning process to improve the efficiency of data collection. The physics-constrained dictionary learning approach has been demonstrated with 1D signals where sampling time stamps are optimized. In 2D images, the optimal locations of pixels to sample are determined. The high-fidelity information can be reconstructed with limited measurements at designed time stamps for 1D signals and pixel locations for 2D images. When monitoring the surface temperature of the builds in AM processes with infrared thermal imaging systems, low-resolution pixel values at the designed locations can be used to reconstruct high-resolution images. Based on the recovered images, more accurate 3D temperature distributions can be reconstructed with PBCS.

8.2 Contributions of this dissertation

To address the limitations of conventional sensors and improve the efficiency of data collection in AM process monitoring, a novel sensing framework called physics-based compressive sensing is developed to improve the compression ratio and accuracy of sensing by integrating the conventional compressed sensing with physical knowledge of the phenomena. The PBCS formulations to monitor temperature distribution in the FFF process, the flow velocity field, and the thermal-fluid behaviour of the melt pool in metal AM are developed. A Gaussian process based uncertainty quantification method is introduced to predict and compensate the systematic error of PBCS during the recovery and reconstruction procedures. A constrained orthogonal matching pursuit algorithm and a domain decomposition method are developed to improve the efficiency of PBCS. The sensing performance is further improved with a physics-constrained dictionary learning approach by simultaneously optimizing measurement and basis matrices with the constraints of sensor types and placement. With the physics-constrained dictionary learning approach, the amount of data collection can be further reduced with the optimized sensor placement.

8.3 Future work

In this dissertation, the temperature monitoring of AM process is demonstrated with single-track printing. In future work, the PBCS approach for complete fabrication process will be developed. The PBCS approach based on the heat transfer model in CHAPTER 3 can be used to monitor the temperature distribution of the complete build domain in metal AM processes. The PBCS formulation based on the thermofluid model will be used to monitor the melt pool with the size of micrometers in the complete fabrication process with the moving heating source. The reliability of physical models is critical for PBCS reconstruction performance. Simplification and approximation are always involved with modelling. Therefore, the thermofluid model will be

extended to include more detailed physics such as vaporization due to high laser power and the effects of the melt pool dynamics on surface roughness. The computational cost of PBCS recovery and reconstruction is expensive for in-situ process monitoring, where physical quantities to be monitored are changing rapidly. The computation of the 3D temperature distribution and melt flow from the thermofluid model is particularly expensive, because the mesh density of the melt pool needs to be high, which results in the large size of matrices used in recovery and reconstruction. Therefore, approaches to further reduce the computational cost for in-situ process monitoring need to be developed. The obvious way is to accelerate the computation using parallel computers, as the developed PBCS formulation and models can be easily parallelized.

As a part of the developed physics-constrained dictionary learning method, a discrete Bayesian optimization (dBO) method was used to solve the combinatorial optimization problem. The dBO searching method is based on GPR surrogate models so as to reduce the variability of searching results. However, the computational cost of the GPR model is still high. Other surrogate models such as random forests, support vector machines, and gradient-enhanced kriging can be used to replace the GPR model to potentially improve the efficiency of the physics-constrained dictionary learning method. The K-SVD was used as the training algorithm in the physics-constrained dictionary learning method. The performance of reconstruction depends on the choices of the initial basis matrix and the reconstruction algorithm, because the K-SVD can only find the local optima. The initial basis matrix in K-SVD can be designed to capture more correlations in the training data. Instead of using the K-SVD, other dictionary learning algorithms can also be applied. For instance, the online dictionary learning is more efficient than the K-SVD algorithm when monitoring real-time systems, because it does not need to store and access the entire dataset.

The physics-constrained dictionary learning method developed here can also be applied for

feature detection and machine health diagnostics based on the signals with different types of defects as the training datasets. For instance, collected optical images to monitor the melt pool in metal AM can be used to identify defects such as powder spatter, gas and process induced porosity, and irregular shapes of the melt pool. Based on different types of defects, the physics-constrained dictionary learning method can be used to classify defects for the AM process monitoring by incorporating modified K-SVD algorithms for classification such as discriminative K-SVD [124] and label consistent K-SVD [125]. The objective function of the physics-constrained dictionary learning method will also be used to minimize the classification error. In addition to the measurement and basis matrices which are used for CS recovery, a classifier matrix can also be optimized to classify different defects in terms of the recovered coefficient vector.

The physics-constrained dictionary learning method can be extended for higher dimensional signals such as videos. In CHAPTER 6 and CHAPTER 7, the physics-constrained dictionary learning method has been used to reduce the number of data collection in temporal and spatial domains respectively. For higher dimensional signals in both of temporal and spatial domains, the measurement matrices to indicate time stamps and locations of data collection can be optimized simultaneously. To monitor the AM processes, high-speed optical cameras are usually used. There is a trade-off between the spatial and temporal resolutions in the use of high-speed cameras. For example, to capture videos with high-spatial resolution, the temporal resolution of the measurements will be low. The physics-constrained dictionary learning method can be applied to increase the resolution of data collection. Based on the measurements with either low spatial or low temporal resolutions, high-fidelity information can be reconstructed. To optimize the measurement matrices in spatial and temporal domains respectively, physical constraints can be introduced to build the relation between the location of moving heat source and the time stamp,

which can be obtained from machine settings or G-code. To monitor the local region such as the melt pool in metal AM, a coaxial monitoring system with digital camera can be used [126], where the camera moves with the laser source. As the relative position between the camera and the melt pool is fixed, the location of the camera is not important for PBCS. However, the moving direction of the camera will be used as an additional physical constraint to improve the reconstruction accuracy, because the shape of the melt pool will be changed if the moving direction of the heat source changes. Based on different moving directions, collected images can be classified into different groups. Applying the physics-constrained dictionary learning method for images in the same group after classification can further improve the reconstruction accuracy.

As further extensions, the physics-constrained dictionary learning method can also be applied to design the sensor deployment in large-scale sensor networks by introducing more general constraints such as the communication condition between sensors and coverage of sensors. This will open the door of broader applications of the new sensing approaches developed in this dissertation.

REFERENCES

[1] Baumann, F., & Roller, D. (2016). Vision based error detection for 3D printing processes. In *MATEC web of conferences* (Vol. 59). EDP Sciences.

[2] Liu, S., & Shin, Y. C. (2019). Additive manufacturing of Ti6Al4V alloy: A review. *Materials & Design*, 164, 107552.

[3] DebRoy, T., Wei, H. L., Zuback, J. S., Mukherjee, T., Elmer, J. W., Milewski, J. O., Beese A.M., Wilson-Heid A., De A., & Zhang, W. (2018). Additive manufacturing of metallic components–process, structure and properties. *Progress in Materials Science*, 92, 112-224.

[4] Liu, D. & Wang, Y. (2019) Mesoscale multi-physics simulation of rapid solidification of Ti-6Al-4V alloy. *Additive Manufacturing*, 25: 551-562.

[5] Everton, S. K., Hirsch, M., Stravroulakis, P., Leach, R. K., & Clare, A. T. (2016). Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. *Materials & Design*, *95*, 431-445.

[6] Tapia, G., & Elwany, A. (2014). A review on process monitoring and control in metal-based additive manufacturing. *Journal of Manufacturing Science and Engineering*, *136*(6), 060801-1.

[7] Zhang, B., Liu, S., & Shin, Y. C. (2019). In-Process monitoring of porosity during laser additive manufacturing process. *Additive Manufacturing*, *28*, 497-505.

[8] Shamsaei, N., Yadollahi, A., Bian, L., & Thompson, S. M. (2015). An overview of Direct Laser Deposition for additive manufacturing; Part II: Mechanical behavior, process parameter optimization and control. *Additive Manufacturing*, *8*, 12-35.

[9] Egan, D. S., & Dowling, D. P. (2019). Influence of process parameters on the correlation between in-situ process monitoring data and the mechanical properties of Ti-6Al-4V non-stochastic cellular structures. *Additive Manufacturing*, *30*, 100890.

[10] Dehoff, R. R., Kirka, M. M., Sames, W. J., Bilheux, H., Tremsin, A. S., Lowe, L. E., & Babu, S. S. (2015). Site specific control of crystallographic grain orientation through electron beam additive manufacturing. *Materials Science and Technology*, 31(8), 931-938.

[11] Liu, J., Song, Y., Chen, C., Wang, X., Li, H., Wang, J., Guo, K. & Sun, J. (2020). Effect of scanning speed on the microstructure and mechanical behavior of 316L stainless steel fabricated by selective laser melting. *Materials & Design*, *186*, 108355.

[12] Wang, X., Gong, X., & Chou, K. (2015). Scanning speed effect on mechanical properties of Ti-6Al-4V alloy processed by electron beam additive manufacturing. *Procedia Manufacturing*, *1*, 287-295.

[13] Lo, Y. L., Liu, B. Y., & Tran, H. C. (2019). Optimized hatch space selection in double-scanning track selective laser melting process. *The International Journal of Advanced Manufacturing Technology*, *105*(7-8), 2989-3006.

[14] Candes, E. J., & Tao, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, *52*(12), 5406-5425.

[15] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory*, *52*(4), 1289-1306.

[16] Baraniuk, R. G. (2007). Compressive sensing. *IEEE signal processing magazine*, *24*(4).

[17] Eldar, Y. C., & Kutyniok, G. (Eds.). (2012). *Compressed sensing: theory and applications*. Cambridge University Press.

[18] Lustig, M., Donoho, D., & Pauly, J. M. (2007). Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic resonance in medicine*, *58*(6), 1182-1195

[19] Duarte, M. F., Davenport, M. A., Takhar, D., Laska, J. N., Sun, T., Kelly, K. E., & Baraniuk, R. G. (2008). Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, *25*(2), 83.

[20] Gan, L. (2007). Block compressed sensing of natural images. In *2007 IEEE 15th International conference on digital signal processing* (pp. 403-406).

[21] Haupt, J., Bajwa, W. U., Rabbat, M., & Nowak, R. (2008). Compressed sensing for networked data. *IEEE Signal Processing Magazine*, *25*(2), 92-101

[22] Chen, X., Du, Z., Li, J., Li, X., & Zhang, H. (2014). Compressed sensing based on dictionary learning for extracting impulse components. *Signal Processing*, *96*, 94-109.

[23] Wang, Y., Xiang, J., Mo, Q., & He, S. (2015). Compressed sparse time–frequency feature representation via compressive sensing and its applications in fault diagnosis. *Measurement*, *68*, 70-81.

[24] Wang, H., Ke, Y., Luo, G., & Tang, G. (2015). Compressed sensing of roller bearing fault based on multiple down-sampling strategy. *Measurement Science and Technology*, *27*(2), 025009.

[25]Tang, G., Yang, Q., Wang, H. Q., Luo, G. G., & Ma, J. W. (2015). Sparse classification of rotating machinery faults based on compressive sensing strategy. *Mechatronics*, *31*, 60-67.

[26] Ding, X., & He, Q. (2016). Time–frequency manifold sparse reconstruction: A novel method for bearing fault feature extraction. *Mechanical Systems and Signal Processing*, *80*, 392–413.

[27] Yuan, H., & Lu, C. (2016). Rolling bearing fault diagnosis under fluctuant conditions based on compressed sensing. *Structural Control and Health Monitoring*, *24*(5), e1918.

[28] Liu, C., Wu, X., Mao, J., & Liu, X. (2016). Acoustic emission signal processing for rolling bearing running state assessment using compressive sensing. *Mechanical Systems and Signal Processing*, *91*, 395-406.

[29] Dinwiddie, R. B., Love, L. J., & Rowe, J. C. (2013, May). Real-time process monitoring and temperature mapping of a 3D polymer printing process. *SPIE defense, security, and sensing, International Society for Optics and Photonics*, 87050L.

[30] Rao, P. K., Liu, J. P., Roberson, D., Kong, Z. J., & Williams, C. (2015). Online real-time quality monitoring in additive manufacturing processes using heterogeneous sensors. *Journal of Manufacturing Science and Engineering*, *137*(6), 061007.

[31] Baumann, F., Schön, M., Eichhoff, J., & Roller, D. (2016). Concept Development of a Sensor Array for 3D Printer. *Procedia CIRP*, *51*, 24-31.

[32] Nuchitprasitchai, S., Roggemann, M., & Pearce, J. M. (2017). Factors effecting real-time optical monitoring of fused filament 3D printing. *Progress in Additive Manufacturing*, 1-17.

[33] Wu, H., Yu, Z., & Wang, Y. (2016a). A new approach for online monitoring of additive manufacturing based on acoustic emission. In *ASME 2016 11th International Manufacturing Science and Engineering Conference*, pp. V003T08A013-V003T08A013.

[34] Wu, H., Wang, Y., & Yu, Z. (2016b). In situ monitoring of FDM machine condition via acoustic emission. *The International Journal of Advanced Manufacturing Technology*, **84**(5-8), 1483-1495.

[35] Wu, H., Yu, Z., & Wang, Y. (2017). Real-time FDM machine condition monitoring and diagnosis based on acoustic emission and hidden semi-Markov model. *The International Journal of Advanced Manufacturing Technology*, **90**(5-8), 2027-2036.

[36] Liu, J., Hu, Y., Wu, B., & Wang, Y. (2018). An improved fault diagnosis approach for FDM process with acoustic emission. *Journal of Manufacturing Processes*, *35*, 570-579.

[37] Wu, H., Yu, Z., & Wang, Y. (2019). Experimental study of the process failure diagnosis in additive manufacturing based on acoustic emission. *Measurement*, *136*, 445-453.

[38] Ceruti, A., Liverani, A., & Bombardi, T. (2017). Augmented vision and interactive monitoring in 3D printing process. *International Journal on Interactive Design and Manufacturing*, **11**(2), 385-395.

[39] Cummings, I. T., Bax, M. E., Fuller, I. J., Wachtor, A. J., & Bernardin, J. D. (2017). A Framework for Additive Manufacturing Process Monitoring & Control. In *Topics in Modal Analysis & Testing*, **10**, 137-146. Springer, Cham

[40] Kousiatza, C., & Karalekas, D. (2016). In-situ monitoring of strain and temperature distributions during fused deposition modeling process. *Materials & Design*, **97**, 400-406.

[41] Faes, M., Abbeloos, W., Vogeler, F., Valkenaers, H., Coppens, K., Goedemé, T., & Ferraris, E. (2014). Process monitoring of extrusion based 3D printing via laser scanning. In *PMI 2014 Conference Proceedings*, **6**, 363 – 367.

[42] Craeghs, T., Clijsters, S., Kruth, J. P., Bechmann, F., & Ebert, M. C. (2012). Detection of process failures in layerwise laser melting with optical process monitoring. *Physics Procedia*, *39*, 753-759.

[43] Clijsters, S., Craeghs, T., Buls, S., Kempen, K., & Kruth, J. P. (2014). In situ quality control of the selective laser melting process using a high-speed, real-time melt pool monitoring system. *The International Journal of Advanced Manufacturing Technology*, *75*(5-8), 1089-1101.

[44] Gu, D., & Shen, Y. (2008). Processing conditions and microstructural features of porous 316L stainless steel components by DMLS. *Applied Surface Science*, *255*(5), 1880-1887.

[45] Lott, P., Schleifenbaum, H., Meiners, W., Wissenbach, K., Hinke, C., & Bültmann, J. (2011). Design of an optical system for the in situ process monitoring of selective laser melting (SLM). *Physics Procedia*, *12*, 683-690.

[46] Pavlov, M., Doubenskaia, M., & Smurov, I. (2010). Pyrometric analysis of thermal processes in SLM technology. *Physics Procedia*, *5*, 523-531.

[47] Hooper, P. A. (2018). Melt pool temperature and cooling rates in laser powder bed fusion. *Additive Manufacturing*, *22*, 548-559

[48] Dinwiddie, R. B., Dehoff, R. R., Lloyd, P. D., Lowe, L. E., & Ulrich, J. B. (2013, May). Thermographic in-situ process monitoring of the electron-beam melting technology used in additive manufacturing. In *Thermosense: thermal infrared applications XXXV* (Vol. 8705, p. 87050K). International Society for Optics and Photonics.

[49] Islam, M., Purtonen, T., Piili, H., Salminen, A., & Nyrhilä, O. (2013). Temperature profile and imaging analysis of laser additive manufacturing of stainless steel. *Physics Procedia*, *41*, 835-842.

[50] Schwerdtfeger, J., Singer, R. F., & Körner, C. (2012). In situ flaw detection by IR-imaging during electron beam melting. *Rapid Prototyping Journal*, *18*(4), 259-263.

[51] Rodriguez, E., Mireles, J., Terrazas, C. A., Espalin, D., Perez, M. A., & Wicker, R. B. (2015). Approximation of absolute surface temperature measurements of powder bed fusion additive manufacturing technology using in situ infrared thermography. *Additive Manufacturing*, *5*, 31-39.

[52] Rodriguez, E., Medina, F., Espalin, D., Terrazas, C., Muse, D., Henry, C., MacDonald, E. & Wicker, R. B. (2012). Integration of a thermal imaging feedback control system in electron beam melting. *WM Keck Center for 3D Innovation, University of Texas at El Paso*, 945-961.

[53] Karnati, S., Matta, N., Sparks, T., & Liou, F. (2013). Vision-based process monitoring for laser metal deposition processes. In *Proceedings Solid Freeform Fabrication symposium* (pp. 88-94).

[54] Krauss, H., Eschey, C., & Zaeh, M. (2012, August). Thermography for monitoring the selective laser melting process. In *Proceedings of the solid freeform fabrication symposium* (pp. 999-1014).

[55] Doubenskaia, M., Pavlov, M., & Chivel, Y. (2010). Optical system for on-line monitoring and temperature control in selective laser melting technology. In *Key engineering materials* (Vol. 437, pp. 458-461). Trans Tech Publications Ltd.

[56] Khanzadeh, M., Chowdhury, S., Tschopp, M. A., Doude, H. R., Marufuzzaman, M., & Bian, L. (2019). In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes. *IISE Transactions*, *51*(5), 437-455.

[57] Lane, B., Moylan, S., Whitenton, E. P., & Ma, L. (2016). Thermographic measurements of the commercial laser powder bed fusion process at NIST. *Rapid prototyping journal*, *22*(5), 778-787.

[58] Guo, Q., Zhao, C., Qu, M., Xiong, L., Hojjatzadeh, S. M. H., Escano, L. I., Parab, N.D., Fezzaa, K., Sun, T. & Chen, L. (2020). In-situ full-field mapping of melt flow dynamics in laser metal additive manufacturing. *Additive Manufacturing*, *31*, 100939.

[59] Guo, Q., Zhao, C., Qu, M., Xiong, L., Escano, L. I., Hojjatzadeh, S. M. H., Parab, N.D., Fezzaa, K., Everhart, W., Sun, T. & Chen, L. (2019). In-situ characterization and quantification of melt pool variation under constant input energy density in laser powder bed fusion additive manufacturing process. *Additive Manufacturing*, *28*, 600-609.

[60] Hojjatzadeh, S. M. H., Parab, N. D., Yan, W., Guo, Q., Xiong, L., Zhao, C., Qu, M., Escano, L.I., Xiao, X., Fezzaa, K. & Everhart, W. (2019). Pore elimination mechanisms during 3D printing of metals. *Nature communications*, *10*(1), 1-8.

[61] Wolff, S. J., Wu, H., Parab, N., Zhao, C., Ehmann, K. F., Sun, T., & Cao, J. (2019). In-situ high-speed X-ray imaging of piezo-driven directed energy deposition additive manufacturing. *Scientific reports*, *9*(1), 1-14.

[62] Zhao, C., Fezzaa, K., Cunningham, R. W., Wen, H., De Carlo, F., Chen, L., Rollett, A.D. & Sun, T. (2017). Real-time monitoring of laser powder bed fusion process using high-speed X-ray imaging and diffraction. *Scientific reports*, *7*(1), 1-11.

[63] Ozisik, M. N. (2000). *Inverse heat transfer: fundamentals and applications*. CRC Press.

[64] Park, H. M., & Yoon, T. Y. (2000). Solution of the inverse radiation problem using a conjugate gradient method. *International journal of heat and mass transfer*, *43*(10), 1767-1776.

[65] Chantasiriwan, S. (1999). Inverse heat conduction problem of determining time-dependent heat transfer coefficient. *International Journal of Heat and Mass Transfer*, *42*(23), 4275-4285.

[66] Su, J., & Hewitt, G. F. (2004). Inverse heat conduction problem of estimating time-varying heat transfer coefficient. *Numerical Heat Transfer, Part A: Applications*, *45*(8), 777-789.

[67] Huang, C. H., Chiang, C. C., & Chen, H. M. (1998). Shape identification problem in estimating geometry of multiple cavities. *Journal of thermophysics and heat transfer*, *12*(2), 270-277.

[68] Huang, C. H., Ozisik, M. N., & Sawaf, B. (1992). Conjugate gradient method for determining unknown contact conductance during metal casting. *International Journal of Heat and Mass Transfer*, *35*(7), 1779-1786.

[69] Liu, F. B. (2008). A modified genetic algorithm for solving the inverse heat transfer problem of estimating plan heat source. *International Journal of Heat and Mass Transfer*, *51*(15-16), 3745-3752.

[70] Xue, Y., Zhai, Z. J., & Chen, Q. (2013). Inverse prediction and optimization of flow control conditions for confined spaces using a CFD-based genetic algorithm. *Building and Environment*, *64*, 77-84.

[71] Obayashi, S., & Takanashi, S. (1996). Genetic optimization of target pressure distributions for inverse design methods. *AIAA journal*, *34*(5), 881-886.

[72] Goto, A., Nohmi, M., Sakurai, T., & Sogawa, Y. (2002). Hydrodynamic design system for pumps based on 3-D CAD, CFD, and inverse design method. *J. Fluids Eng.*, *124*(2), 329-335.

[73] Griaznov, V., Trigui, N., Affes, H., & Smith, D. (1999). CFD based shape optimization of IC engine. *Oil & Gas Science and Technology*, *54*(2), 297-307.

[74] O¨ ksu¨ z, O. Z., & Akmandor, I. B. S. (2008, January). Axial turbine blade aerodynamic optimization using a novel multi-level genetic algorithm. In *Turbo Expo: Power for Land, Sea, and Air* (Vol. 43161, pp. 2361-2374).

[75] Waeytens, J., Chatellier, P., & Bourquin, F. (2015). Inverse computational fluid dynamics: Influence of discretization and model errors on flows in water network including junctions. *Journal of Fluids Engineering*, *137*(9).

[76] Salloum, M., Fabian, N. D., Hensinger, D. M., Lee, J., Allendorf, E. M., Bhagatwala, A., Blaylock, M. L., Chen, J. H., Templeton, J. A., and Tezaur, I. (2018). Optimal compressed sensing and reconstruction of unstructured mesh datasets. *Data Science and Engineering*, *3*(1), 1-23.

[77] Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., & Solenthaler, B. (2019, May). Deep fluids: A generative network for parameterized fluid simulations. In *Computer Graphics Forum* (Vol. 38, No. 2, pp. 59-70).

[78] Arigovindan, M., Suhling, M., Jansen, C., Hunziker, P., & Unser, M. (2006). Full motion and flow field recovery from echo Doppler data. *IEEE Transactions on Medical Imaging*, *26*(1), 31-45.

[79] Wang, J. X., & Xiao, H. (2016). Data-driven CFD modeling of turbulent flows through complex structures. *International Journal of Heat and Fluid Flow*, *62*, 138-149.

[80] Duraisamy, K., & Durbin, P. (2014). Transition modeling using data driven approaches. In *CTR Summer Program* (p. 427).

[81] Singh, A. P., Medida, S., & Duraisamy, K. (2017). Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA journal*, *55*(7), 2215-2227.

[82] Kreutz-Delgado, K., Murray, J. F., Rao, B. D., Engan, K., Lee, T. W., & Sejnowski, T. J. (2003). Dictionary learning algorithms for sparse representation. *Neural computation*, *15*(2), 349-396.

[83] Engan, K., Aase, S. O., & Husoy, J. H. (1999, March). Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)* (Vol. 5, pp. 2443-2446). IEEE.

[84] Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, *54*(11), 4311-4322.

[85] Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, *11*(Jan), 19-60.

[86] Chen, X., Du, Z., Li, J., Li, X., & Zhang, H. (2014). Compressed sensing based on dictionary learning for extracting impulse components. *Signal Processing*, *96*, 94-109.

[87] Ding, X., & He, Q. (2016). Time–frequency manifold sparse reconstruction: A novel method for bearing fault feature extraction. *Mechanical Systems and Signal Processing*, *80*, 392–413.

[88] Guo, L., Gao, H., Li, J., Huang, H., & Zhang, X. (2015). Machinery vibration signal denoising based on learned dictionary and sparse representation. In *Journal of Physics: Conference Series* (Vol. 628, No. 1, p. 012124).

[89] Zhou, H., Chen, J., Dong, G., & Wang, R. (2016). Detection and diagnosis of bearing faults using shift-invariant dictionary learning and hidden Markov model. *Mechanical Systems and Signal Processing*, *72*, 65-79.

[90] Lorintiu, O., Liebgott, H., Alessandrini, M., Bernard, O., & Friboulet, D. (2015). Compressed sensing reconstruction of 3D ultrasound data using dictionary learning and line-wise subsampling. *IEEE transactions on medical imaging*, *34*(12), 2467-2477.

[91] Huang, Y., Paisley, J., Lin, Q., Ding, X., Fu, X., & Zhang, X. P. (2014). Bayesian nonparametric dictionary learning for compressed sensing MRI. *IEEE Transactions on Image Processing*, *23*(12), 5007-5019.

[92] Ravishankar, S., & Bresler, Y. (2010). MR image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE transactions on medical imaging*, *30*(5), 1028-1041.

[93] Ravishankar, S., & Bresler, Y. (2013, April). Sparsifying transform learning for compressed sensing MRI. In *2013 IEEE 10th International Symposium on Biomedical Imaging* (pp. 17-20). IEEE.

[94] Zhan, Z., Cai, J. F., Guo, D., Liu, Y., Chen, Z., & Qu, X. (2015). Fast multiclass dictionaries learning with geometrical directions in MRI reconstruction. *IEEE Transactions on biomedical engineering*, *63*(9), 1850-1861.

[95] Song, Y., Yang, G., Xie, H., Zhang, D., & Xingming, S. (2017). Residual domain dictionary learning for compressed sensing video recovery. *Multimedia Tools and Applications*, *76*(7), 10083-10096.

[96] Polania, L. F., & Barner, K. E. (2013, August). Multi-scale dictionary learning for compressive sensing ECG. In *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)* (pp. 36-41). IEEE.

[97] Dong, W., Li, X., Zhang, L., & Shi, G. (2011, June). Sparsity-based image denoising via dictionary learning and structural clustering. In *CVPR 2011* (pp. 457-464). IEEE.

[98] Zhang, X., Feng, X., Wang, W., & Liu, G. (2013). Image denoising via 2D dictionary learning and adaptive hard thresholding. *Pattern Recognition Letters*, *34*(16), 2110-2117.

[99] Zhou, M., Chen, H., Paisley, J., Ren, L., Li, L., Xing, Z., ... & Carin, L. (2011). Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, *21*(1), 130-144.

[100] Duarte-Carvajalino, J. M., & Sapiro, G. (2009). Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. IEEE Transactions on Image Processing, 18(7), 1395-1408.

[101] Bai, H., Li, G., Li, S., Li, Q., Jiang, Q., & Chang, L. (2015). Alternating optimization of sensing matrix and sparsifying dictionary for compressed sensing. *IEEE Transactions on Signal Processing*, *63*(6), 1581-1594.

[102] Lu, Y., & Wang, Y. (2018). Monitoring temperature in additive manufacturing with physics-based compressive sensing. *Journal of manufacturing systems*, *48*, 60-70.

[103] Lu, Y., & Wang, Y. (2018). An Efficient Transient Temperature Monitoring of Fused Filament Fabrication Process with Physics Based Compressive Sensing. *IISE Transactions*, (in press), 1-30.

[104] Tropp, J. A., & Gilbert, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, ***53***(12), 4655-4666.

[105] Abolghasemi, V., Ferdowsi, S., Makkiabadi, B., & Sanei, S. (2010, August). On optimization of the measurement matrix for compressive sensing. In *Signal Processing Conference, 2010 18th European* (pp. 427-431). IEEE.

[106] Huang, A., Guan, G., Wan, Q., & Mehbodniya, A. (2011). A block orthogonal matching pursuit algorithm based on sensing dictionary. *International Journal of Physical Sciences*, ***6***(5), 992-999.

[107] Wang, J., Kwon, S., & Shim, B. (2012). Generalized orthogonal matching pursuit. *IEEE Transactions on signal processing*, ***60***(12), 6202.

[108] Dai, W., & Milenkovic, O. (2009). Subspace pursuit for compressive sensing signal reconstruction. *IEEE transactions on Information Theory*, ***55***(5), 2230-2249.

[109] Lu, Y., & Wang, Y. (2019). An Improvement of Physics Based Compressive Sensing With Domain Decomposition to Monitor Temperature in Fused Filament Fabrication Process. In *ASME 14th International Manufacturing Science and Engineering Conference*, Erie, Pennsylvania, **58745**, p. V001T01A032

[110] Chen, S. S., Donoho, D. L., & Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM review*, *43*(1), 129-159.

[111] FEniCS, Software package, 2011. https://fenicsproject.org/.

[112] Lu, Y., & Wang, Y. (2020). Physics Based Compressive Sensing Approach to Monitor Turbulent Flow. *AIAA journal*, *58*(8), 3299-3307.

[113] Launder, B. E., & Spalding, D. B. (1983). The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion* (pp. 96-116). Pergamon.

[114] Kuzmin, D., Mierka, O., & Turek, S. (2007). On the implementation of the κ-ε turbulence model in incompressible flow solvers based on a finite element discretisation. *International Journal of Computing Science and Mathematics*, *1*(2-4), 193-206.

[115] Driver, D. M., & Seegmiller, H. L. (1985). Features of a reattaching turbulent shear layer in divergent channelflow. *AIAA journal*, *23*(2), 163-171.

[116] Huang, R., Luo, X., Ji, B., & Ji, Q. (2017). Turbulent flows over a backward facing step simulated using a modified partially averaged Navier–Stokes model. *Journal of Fluids Engineering*, *139*(4).

[117] Thangam, S., & Speziale, C. G. (1992). Turbulent flow past a backward-facing step-A critical evaluation of two-equation models. *AIAA journal*, *30*(5), 1314-1320.

[118] Saad, Y. (1981). Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of computation*, *37*(155), 105-126.

[119] Mardal, K. A., & Winther, R. (2011). Preconditioning discretizations of systems of partial differential equations. *Numerical Linear Algebra with Applications, 18*(1), 1-40.

[120] Ranieri, J., Chebira, A., & Vetterli, M. (2014). Near-optimal sensor placement for linear inverse problems. *IEEE Transactions on signal* processing, 62(5), 1135-1146.

[121] Wang, B., Lei, Y., Li, N., & Li, N. (2018). A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*.

[122] Wang Y. "Design of trustworthy cyber-physical systems with discrete Bayesian optimization." *Proceedings of 2020 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE2020), August 16-19, 2020*, Paper No. DETC2020-22661

[123] Zhang J., Yao X., Liu M., and Wang Y. "A Bayesian discrete optimization for permutation based combinatorial problems." *Proceedings of IEEE Symposium Series on Computational Intelligence, December 6-9, 2019*, pp.874-881.

[124] Zhang, Q., & Li, B. (2010, June). Discriminative K-SVD for dictionary learning in face recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2691-2698). IEEE.

[125] Jiang, Z., Lin, Z., & Davis, L. S. (2013). Label consistent K-SVD: Learning a discriminative dictionary for recognition. IEEE transactions on pattern analysis and machine intelligence, 35(11), 2651-2664.

[126] Gökhan Demir, A., De Giorgi, C., & Previtali, B. (2018). Design and implementation of a multisensor coaxial monitoring system with correction strategies for selective laser melting of a maraging steel. *Journal of Manufacturing Science and Engineering*, *140*(4).